



Universidad de Carabobo

Facultad Experimental De Ciencias y Tecnología

Departamento De Computación

Redes de Computadoras I - CAO603

Valencia - Venezuela

Optimización

Autores:

Omar Gonzalez 27244029

Introducción

Las redes de área local, generalmente llamadas LAN (Local Area Networks), son redes de propiedad privada que operan dentro de un solo edificio, como una casa, oficina o fábrica. Las redes LAN se utilizan ampliamente para conectar computadoras personales y electrodomésticos con el fin de compartir recursos e intercambiar información. En estos sistemas, cada computadora tiene un módem y una antena que utiliza para comunicarse con otras computadoras.

La mayoría utilizan cables de cobre, pero algunas usan fibra óptica. Las redes LAN tienen restricciones en cuanto a su tamaño, lo cual significa que el tiempo de transmisión en el peor de los casos es limitado y se sabe de antemano. Conocer estos límites facilita la tarea del diseño de los protocolos de red. Por lo general las redes LAN inalámbricas operan a velocidades que van de los 100 Mbps hasta un 1 Gbps.

La topología de muchas redes LAN alámbricas está basada en los enlaces de punto a punto. El estándar IEEE 802.3, comúnmente conocido como Ethernet, es hasta ahora el tipo más común de LAN alámbrica.

Cada computadora se comunica mediante el protocolo Ethernet y se conecta a una caja conocida como switch con un enlace de punto a punto. De aquí que tenga ese nombre. Un switch tiene varios puertos, cada uno de los cuales se puede conectar a una computadora. El trabajo del switch es transmitir paquetes entre las computadoras conectadas a él, y utiliza la dirección en cada paquete para determinar a qué computadora se lo debe enviar. Para crear redes LAN más grandes se pueden conectar switches entre sí mediante sus puertos.

Objetivos

Objetivo general

Implementar un programa que sea capaz de optimizar la cantidad de cable utilizado en una conexión FTTB mediante la utilización de un Switch

Objetivos Específicos

- Crear un repositorio en github para que sea posible mantener una programación organizada mediante un manejador de versiones.
- Obtener los conocimientos necesarios del lenguaje utilizado para la implementación, como la sintaxis, funciones y librerías a utilizar.
- Implementar los algoritmos de interés tomando en cuenta todos los factores necesarios para lograr un buen funcionamiento, detallar y comentar el código de forma que el código sea lo mas optimo y legible posible.
- Agregar las validaciones necesarias para lograr una robustez óptima en el programa.
- Crear una vista sencilla de utilizar pero completa e intuitiva que permita una buena experiencia de usuario.

Marco teórico

FTTB significa "Fiber to the Building", que se traduce como "Fibra hasta el edificio".

Es un tipo de tecnología de conexión a Internet que utiliza cables de fibra óptica para proporcionar acceso a Internet de alta velocidad a edificios residenciales o comerciales.

En comparación con otras tecnologías de conexión a Internet, como DSL o cable coaxial, FTTB ofrece velocidades de descarga y carga más rápidas, así como un ancho de banda más amplio. Esto lo hace ideal para aplicaciones que requieren un uso intensivo de datos, como el streaming de video, los juegos en línea y la descarga de archivos grandes.

FTTB también es más confiable que otras tecnologías de conexión a Internet, ya que es menos susceptible a interferencias y cortes de servicio. Esto lo convierte en una buena opción para negocios o personas que necesitan una conexión a Internet estable y confiable.

Las ventajas de FTTB:

- Velocidades de descarga y carga más rápidas
- Mayor ancho de banda
- Conexión más confiable
- Menos susceptible a interferencias
- Ideal para aplicaciones que requieren un uso intensivo de datos

FTTB es una excelente opción. Sin embargo, es importante verificar si FTTB está disponible en tu área, ya que aún no se ha implementado en todas partes.

Switch

Un switch de red es un dispositivo de interconexión utilizado para conectar equipos en red formando lo que se conoce como una red de área local (LAN) y cuyas especificaciones técnicas siguen el estándar conocido como Ethernet (o técnicamente IEEE 802.3).

La función básica de un switch es la de unir o conectar dispositivos en red. Para ello, utiliza una tabla de direcciones MAC, que es una tabla que almacena la dirección MAC de cada dispositivo conectado al switch. Cuando un dispositivo envía un paquete de datos a otro dispositivo, el switch consulta su tabla de direcciones MAC para determinar el puerto al que está conectado el dispositivo de destino. El switch luego reenvía el paquete de datos al puerto correspondiente.

Los switches se utilizan en una amplia gama de aplicaciones, incluyendo:

- **En redes domésticas**, para conectar dispositivos como computadoras, impresoras y televisores.
- **En redes empresariales**, para conectar servidores, estaciones de trabajo y dispositivos periféricos.
- **En redes inalámbricas**, para conectar dispositivos inalámbricos como computadoras portátiles, teléfonos inteligentes y tabletas.

Los switches ofrecen una serie de ventajas sobre otros dispositivos de interconexión, como los hubs. Los switches ofrecen un mejor rendimiento, ya que no comparten el ancho de banda de la red entre todos los dispositivos conectados. También ofrecen una mejor seguridad, ya que pueden aislar los datos de los dispositivos no autorizados.

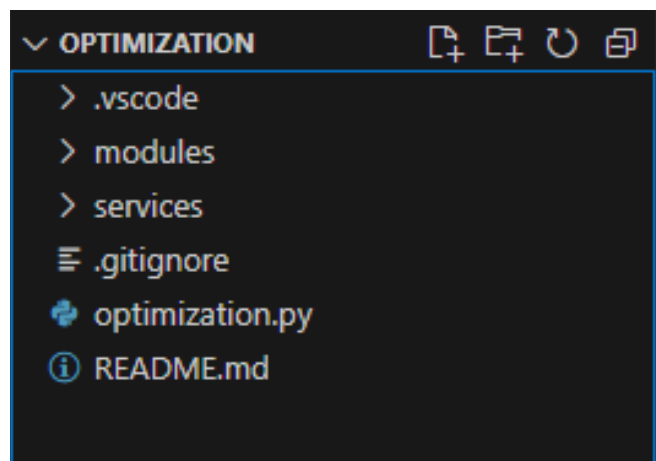
Procedimiento

El procedimiento inicia con la creación de un repositorio en github con una organización de carpetas adecuada para comenzar el desarrollo, seguido de esto se procede a realizar una capacitación sobre el lenguaje a utilizar el cual es python, luego de esto cuando ya se posee un conocimiento suficiente para llevar a cabo la implementación se procede a comenzar con la programación.

Estructura de carpeta

En la estructura de carpeta contaremos con:

- **Archivo principal** “optimization.py” el cual se encargará de crear la interfaz del programa, este se encargará de llamar los servicios necesarios.
- **Carpeta modules** contendrá el módulo principal donde se importaran todos los servicios de la aplicación para que sean accedido con facilidad.
- **Carpeta services** contendrá todos los servicios que a la vez contiene las funciones necesarias para resolver una tarea en específico.



Algoritmo:

1. Variables Globales:

```
optimization.py  points_service.py X
services > points_service.py > calculateDistanceByPoint
You, 48 minutes ago | 1 author (You)
1  import math
2
3  POINT_ZERO = (0,0) #Punto original
4  INFINITY = 99999999999999999999 #Infinito
5
```

2. Función para calcular la distancia:

Calcula la distancia total de la conexiones desde un punto de origen a todos los puntos del destino

```
5
6  """
7  Funcion que calcula la distancia desde un punto de origen hasta multiples puntos de llegada
8  point: un punto (X,Y)
9  pointsList: una lista de puntos (X,Y)
10 """
11 def calculateDistance(point: tuple, pointsList: list):
12     distance = 0
13     #Se calcula la distancia del punto de origen hasta los multiples puntos de llegada
14     for actual in pointsList:
15         vector = math.pow((point[0] - actual[0]), 2) + math.pow((point[1] - actual[1]), 2)
16         distance += math.sqrt(vector)
17     return int(distance)
18
```

3. Función calcular la distancia por puntos:

Consiste en después de sugerir los puntos para optimizar la instalación, calcula la distancia de cada punto sugerido hasta conseguir la más óptima. En el proceso, cada nuevo punto sugerido (que sería donde se instalaría el switch) se le calcula la lista de puntos que irían conectado a él

```

"""
Funcion que calcula la distancia desde los puntos de origen propuesto hasta multiples puntos de llegada
retorna el punto con menor distancia
points: lista de puntos (X,Y) propuestp para la optimizacion
pointsList: una lista de puntos (X,Y) ingresada por el usuario
"""
def calculateDistanceByPoint(points: list, pointsList: list):
    distance = 0
    result: tuple = (POINT_ZERO, INFINITY)
    i = 1
    #Se recorren los puntos propuesto para la optimizacion
    for actual in points:
        #Se consiguen los puntos conectados al hub y los puntos conectados al central principal
        newList = generateNewPointList(pointsList, i)
        listPointNew = newList[1]
        #Se calcula la distancia
        if len(listPointNew) > 0:
            listPointNew.insert(0, POINT_ZERO)
            distance = (calculateDistance(actual, listPointNew), calculateDistance(actual, listPointNew) + calculateDistance
        else:
            distance = calculateDistance(POINT_ZERO, pointsList)
        #Se guarda la menor
        if distance < result[1]:
            result = (actual, distance)
        i+=1
    return result

```

4. Función para generar la nueva lista de puntos para el switch propuesto

Para generar esta nueva lista de puntos para el switch propuesto, se verifica que el tipo, el tipo indicará el rango que ese switch puede cubrir, si no está dentro del rango no se guarda, en caso contrario se guarda ese punto

```

46
47
48 """
49 Funcion que calcula las conexiones para el punto propuesto y para el punto inicial
50 pointsList: una lista de puntos (X,Y) ingresada por el usuario
51 type: tipo del cuadrante
52 """
53 def generateNewPointList(pointsList: list, type: int):
54     rangeX: tuple
55     rangeY: tuple
56     if type == 1:
57         rangeX = (0, INFINITY)
58         rangeY = (0, INFINITY)
59 > elif type == 2: ...
61 > elif type == 3: ...
64 > elif type == 4: ...
67 elif type == 5:
68     rangeX = (-1*INFINITY, INFINITY)
69     rangeY = (0, INFINITY)
70 > elif type == 6: ...
73 > elif type == 7: ...
76 > elif type == 8: ...
79
80 listPoint0 = []
81 listPointNew = []
82 for point in pointsList:
83     if point[0] >= rangeX[0] and point[0] <= rangeX[1] and point[1] >= rangeY[0] and point[1] <= rangeY[1]:
84         listPointNew.insert(0, point)
85     else:
86         listPoint0.insert(0, point)
87 > return (listPoint0, listPointNew)
88

```

5. Función Buscar nuevos puntos

Esta función se encarga de buscar los 8 puntos posible donde se podría colocar el switch para optimizar la instalación.

```
107
108 """
109 Se busca los nuevos puntos posible para instalar el switch, para esto se analiza los 8 caso posibles
110 """
111 def searchNewPoint(pointsList: list):
112     quadrantXPositiveYPositive = newPoint(pointsList, (0, INFINITY), (0, INFINITY))
113     quadrantXPositiveYNegative = newPoint(pointsList, (0, INFINITY), (-1 * INFINITY, -1))
114     quadrantXNegativeYNegative = newPoint(pointsList, (-1 * INFINITY, -1), (-1 * INFINITY, -1))
115     quadrantXNegativeYPositive = newPoint(pointsList, (-1 * INFINITY, -1), (0, INFINITY))
116     quadrantXYPositive = newPoint(pointsList, (-1 * INFINITY, INFINITY), (0, INFINITY))
117     quadrantXYNegative = newPoint(pointsList, (-1 * INFINITY, INFINITY), (-1 * INFINITY, -1))
118     quadrantYXNegative = newPoint(pointsList, (-1 * INFINITY, -1), (-1 * INFINITY, INFINITY))
119     quadrantYXPositive = newPoint(pointsList, (0, INFINITY), (-1 * INFINITY, INFINITY))
120     return [quadrantXPositiveYPositive, quadrantXPositiveYNegative, quadrantXNegativeYNegative, quadrantXNegativeYPositive, quadrantXYPositive, quadrantXYNegative, quadrantYXNegative, quadrantYXPositive]
121
122 def optimizeDistance(pointsList: list):
```

6. Función optimizar distancia

Es la función principal y es la encargada de llamar a las funciones necesaria para obtener el punto donde se colocara el switch y la distancia de cable que se necesitaría.

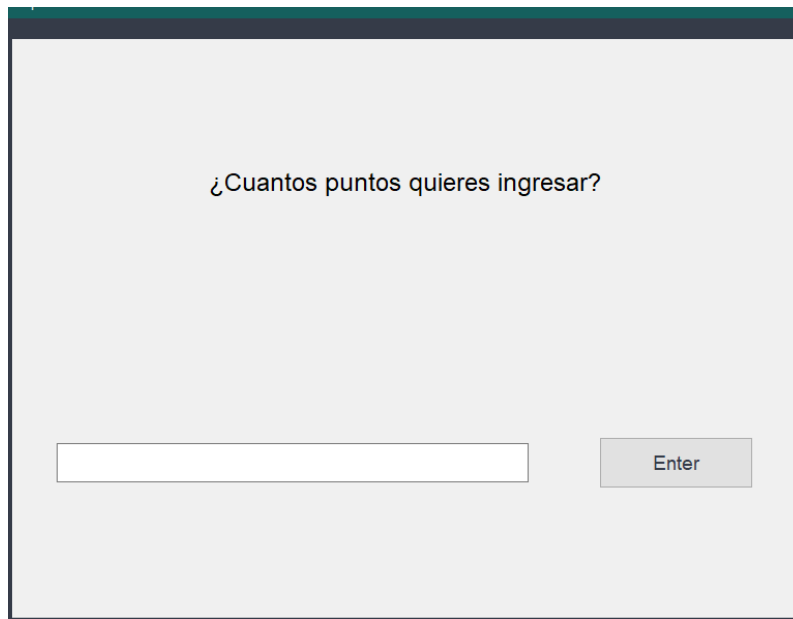
```
121
122 def optimizeDistance(pointsList: list):
123     distanceOriginal = calculateDistance(POINT_ZERO, pointsList)
124     points = searchNewPoint(pointsList)
125     result = calculateDistanceByPoint(points, pointsList)
126     return (result, distanceOriginal)
127
```

Resultados

En los resultados obtenidos procederemos a realizar algunas pruebas y así plasmar el seguimiento y resultado final en cada caso:

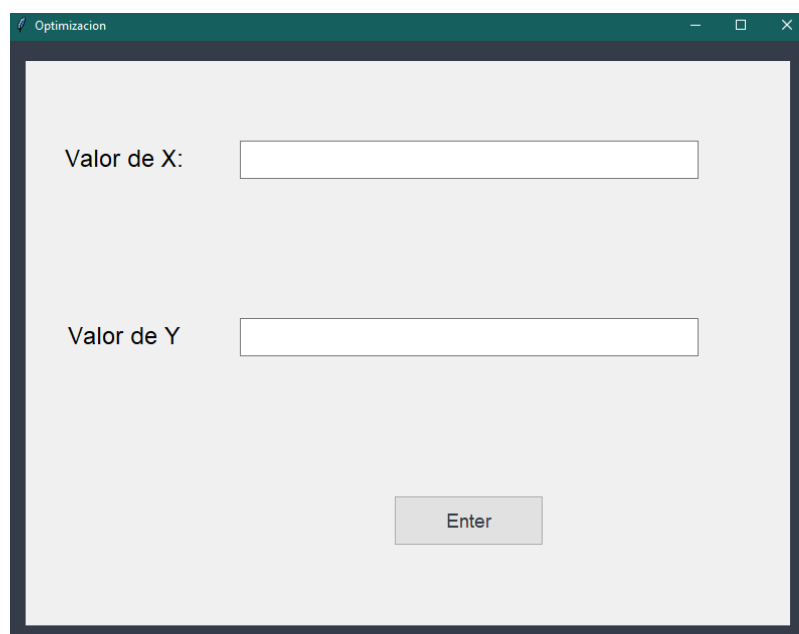
Utilizaremos la cantidad de 4 puntos los cuales serán $(-3,7)$ $(3,7)$ $(-6,7)$ $(6,7)$.

En la primera vista se ingresa la cantidad de puntos a ingresar



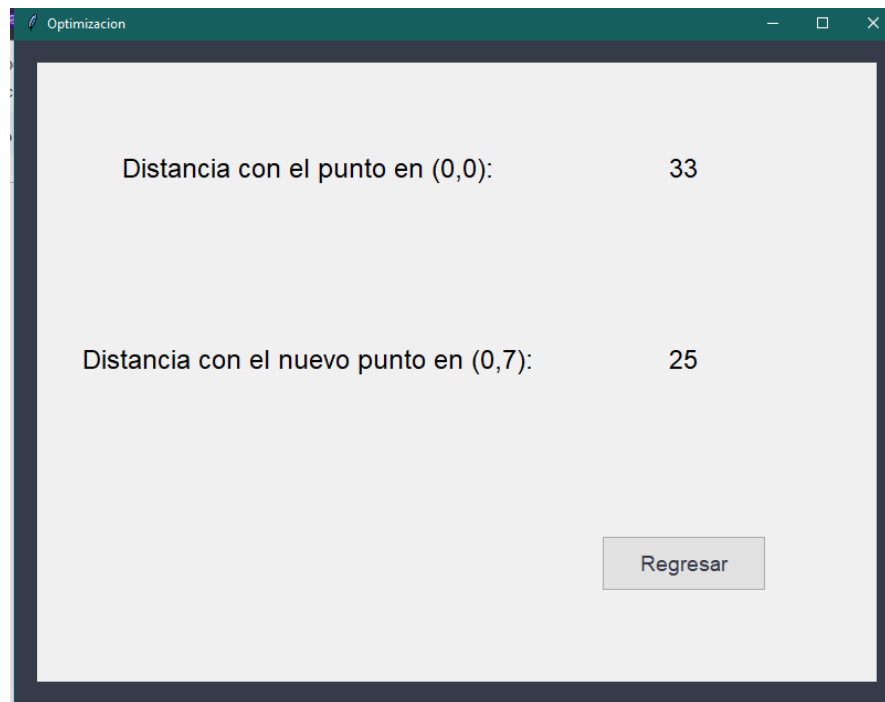
A screenshot of a software window with a light gray background and a dark border. The text "¿Cuántos puntos quieres ingresar?" is centered. Below the text is a white rectangular input field. To the right of the input field is a gray button with the text "Enter".

En la segunda vista se pedirá el punto que se quiere ingresar, después de ingresarlo se repetirá hasta que haya ingresado los n puntos que se definió anteriormente



A screenshot of a software window titled "Optimizacion" with a light gray background and a dark border. It contains two input fields. The first is labeled "Valor de X:" and the second is labeled "Valor de Y:". Below these fields is a gray button with the text "Enter".

Y en la última vista se mostrará la cantidad de cable que se usa sin utilizar un switch, es decir todos los puntos conectados al punto (0,0). También se muestra el punto óptimo donde puede colocar un switch y la cantidad de cable que se usaría en ese punto.



Conclusión

Una vez superadas las fases de preparación, análisis, implementación, maquetación, correcciones, mejoras y evaluación del proyecto, podemos afirmar que tanto los objetivos generales como los específicos fueron cumplidos.

En cuanto a él algoritmos implementado se concluye que analizando el punto óptimo donde instalar un switch en un servicio FFTB se puede reducir de una manera eficaz la cantidad de cable utilizado. Esto sería una ganancia para el proveedor del servicio de internet ya que con una menor cantidad de cable usada, se ahorra más dinero a la compañía, además de la ganancia que se puede obtener a utilizar una estructura ordenada, ya que sería mucho más fácil la modificación o reparación de este sistema

Al concluir este proyecto podemos afirmar que hemos reforzado nuestro conocimiento tanto práctico como teórico de lo importante que es analizar el sistema a proponer ya que de eso dependerá de proveer un buen servicio y ahorrar dinero a la ISP.

Bibliografía

Andrew S. Tanenbaum y David J. Wetherall. Redes de computadoras Quinta edición. Pearson Educación, México, 2012. ISBN: 978-607-32-0817-8. Área: Computación.