

Informe sobre el algoritmo de análisis de tramas IEEE 802.15.4

El algoritmo proporcionado es un programa en C++ diseñado para analizar tramas de datos de la capa de enlace del protocolo IEEE 802.15.4. El objetivo principal del programa es contar y clasificar las tramas recibidas según su longitud, verificar la integridad de las tramas utilizando el checksum, y detectar la presencia de secuencias de escape.

Descripción del algoritmo:

1. Función `analizar_tramas`:

- Esta función recibe una cadena de tramas como entrada y realiza el análisis de cada una de ellas.
- Declara e inicializa las variables necesarias para contabilizar diferentes aspectos de las tramas.
- Utiliza un bucle `while` para iterar a través de la cadena de tramas.

2. Bucle principal:

- Comienza con la declaración de variables locales, como `flag` y `suma`, que se utilizan para el análisis de cada trama.
- Verifica si el primer byte de la trama es el delimitador de trama (`0x7E`). Si no es así, se salta al siguiente byte.

3. Análisis de la trama:

- Si el delimitador de trama es encontrado, se extrae la longitud de la trama y se inicia un bucle para leer cada byte de la trama.
- Durante la lectura de los bytes, se acumulan en una variable `suma` para calcular el checksum y se comprueba si hay secuencias de escape (`0x7D`).
- La longitud de la trama se calcula multiplicando el valor hexadecimal de longitud por 2, ya que cada byte se representa por dos caracteres en la cadena.

4. Verificación de la longitud y el checksum:

- Una vez que se leen todos los bytes de la trama, se verifica si la longitud de la trama es correcta y si el checksum coincide con el valor calculado.

- Se imprime información detallada sobre cada trama, incluida su longitud, contenido, resultado de la verificación del checksum y la detección de secuencias de escape.

5. Actualización de contadores y puntero de posición:

- Se actualizan los contadores de tramas según el resultado de la verificación de la longitud y el checksum.
- El puntero de posición se mueve a la siguiente trama para continuar el análisis.

6. Impresión de estadísticas finales:

- Al finalizar el análisis de todas las tramas, se imprimen las estadísticas finales, que incluyen el número total de tramas, tramas con longitud correcta e incorrecta, tramas con checksum correcto e incorrecto, y el número de tramas que utilizan secuencias de escape.

Para ejecutar el código proporcionado, sigue estos pasos:

1. Guarda el archivo con una extensión "analisis_tramas.cpp".
2. Descarga el archivo "Tramas_802-15-4.log" y asegúrate de que esté en el mismo directorio que el archivo de código fuente ".cpp".
5. Abre una terminal o línea de comandos y navega hasta el directorio donde guardaste el archivo ".cpp".
6. Compila el código escribiendo el siguiente comando en la terminal y presionando Enter:

```
g++ analisis_tramas.cpp -o analisis_tramas
```

Esto compilará el código y generará un archivo ejecutable llamado "analisis_tramas" en el mismo directorio.

7. Una vez compilado, ejecuta el programa escribiendo el siguiente comando en la terminal y presionando Enter:

```
./analisis_tramas
```

Esto ejecutará el programa, que analizará las tramas en el archivo "Tramas_802-15-4.log" y mostrará las estadísticas correspondientes en la terminal.

8. Observa la salida en la terminal para ver las estadísticas del análisis de tramas.

Conclusiones:

El algoritmo proporcionado es capaz de realizar un análisis detallado de las tramas IEEE 802.15.4, identificando errores de longitud, errores de checksum y detectando secuencias de escape. Sin embargo, es importante tener en cuenta que el correcto funcionamiento del algoritmo depende de la precisión y validez de los datos de entrada. Además, el algoritmo podría mejorarse para manejar casos extremos o situaciones inesperadas que puedan surgir en la práctica.