

Memoria del código de ReceiveFrames.c

El código proporcionado es un programa escrito en lenguaje C que utiliza la biblioteca "pcap" para capturar y procesar tramas de red. A continuación, se realizará una memoria detallada del funcionamiento del código.

El programa comienza con la inclusión del archivo de encabezado (.h) que nos permitirá manipular las tramas de red:

- **"pcap.h"**: Es un archivo de encabezado específico del programa que contiene las definiciones y funciones necesarias de la biblioteca "libpcap.lib" (MacOS) o "wpcap.lib" (Windows) para capturar y procesar tramas de red.

A continuación, se define la constante `MESSAGE_INIT_FRAME` con un valor de 14. Esta constante representa el índice inicial del mensaje dentro de la trama Ethernet.

Luego, prototiparemos dos funciones antes del main:

- **`char* getEthernet(void)`**
- **`void packetHandler(u_char *userData, const struct pcap_pkthdr* pkthdr, const unsigned char* packetData)`**

La primera **`char* getEthernet(void)`** se encargará de obtener la interfaz de ethernet que se desea utilizar para ponerse en escucha.

Para ello, mostrara al usuario una lista de interfaces con un numero asignado:

```
1. en0 (No description available)
2. awd10 (No description available)
3. llw0 (No description available)
4. utun0 (No description available)
5. utun1 (No description available)
6. utun2 (No description available)
7. utun3 (No description available)
8. utun4 (No description available)
9. utun5 (No description available)
10. utun7 (No description available)
11. utun8 (No description available)
12. lo0 (No description available)
13. anpi1 (No description available)
14. anpi0 (No description available)
15. en3 (No description available)
16. en4 (No description available)
17. en1 (No description available)
18. en2 (No description available)
19. bridge0 (No description available)
20. gif0 (No description available)
21. stf0 (No description available)
22. ap1 (No description available)
Seleccione el numero de interfez que desea: 1|
```

Por ejemplo, si elegimos la interfaz con el numero 1, la función nos estaría devolviendo su nombre, es decir, en0.

La segunda función **`void packetHandler(u_char *userData, const struct pcap_pkthdr* pkthdr, const unsigned char* packetData)`** se encargara de procesar las tramas de red capturadas

Esta función recibe tres parámetros de entrada:

- **userData**: Puntero a datos adicionales pasados a la función (en este caso, no se utiliza).
- **pkthdr**: Puntero a una estructura pcap_pkthdr que contiene información sobre la trama capturada, como la longitud y la marca de tiempo.
- **packetData**: Puntero a la trama capturada.

Dentro de la función, se verifica si la dirección MAC de destino es de broadcast. Esto se hace verificando si los primeros 6 bytes del paquete recibido (packetData) son todos 0xFF, es decir, que la MAC es FF:FF:FF:FF:FF:FF.

Si se cumple que la MAC es la del broadcast, se verifica si el campo "Ethertype" de la trama es igual a 0x2223. Esto se hace verificando si los bytes 12 y 13 del paquete recibido (packetData) son iguales a 0x22 y 0x23, respectivamente.

Si se cumple de nuevo esta condición, se muestra el mensaje recibido en la trama. Se mostrara la dirección MAC del remitente y los caracteres del mensaje se imprimirán a partir de la posición MESSAGE_INIT_FRAME en adelante.

Por último, pasaremos a explicar un poco lo que hace el main.

Primero, declaramos las variables **errbuf** y **handle**. **errbuf** es un búfer utilizado para almacenar mensajes de error generados por la biblioteca "pcap". **handle** es una variable de tipo **pcap_t** que se utilizará para capturar tramas de red.

Posteriormente encontramos la función **pcap_open_live**, que se utiliza para abrir el dispositivo de red seleccionado y devuelve un puntero al "**handle**" de captura. La función recibe los siguientes parámetros:

- **getEthernet()**: Llama a la función getEthernet, que anteriormente hemos explicado, para obtener el nombre de la interfaz.
- **65536**: Especifica el tamaño máximo de la trama a capturar.
- **1**: Indica el modo promiscuo, que permite capturar todas las tramas de red, incluyendo las que no están destinadas a la interfaz de red seleccionada.
- **1000**: Especifica el tiempo de captura en milisegundos. En este caso, se capturarán tramas durante 1 segundo.
- **errbuf**: Es el búfer donde se almacenará cualquier mensaje de error generado durante la apertura del dispositivo.

Si handle es NULL, se imprime un mensaje de error indicando que no se pudo abrir la interfaz seleccionada.

La función pcap_loop se utiliza para capturar y procesar las tramas de red. Recibe los siguientes parámetros:

- **handle**: El "handle" de captura devuelto por pcap_open_live.
- **0**: Indica que se capturarán todas las tramas de red.
- **packetHandler**: Es el nombre de la función que se llamará para procesar cada trama capturada.
- **NULL**: Puntero a datos adicionales que se pueden pasar a la función **packetHandler**. En este caso, no se utiliza.

Una vez que se completa la captura de tramas, se cierra el "handle" de captura utilizando la función pcap_close.

Ejemplo de funcionamiento:

```
1. en0 (No description available)
2. awdl0 (No description available)
3. llw0 (No description available)
4. utun0 (No description available)
5. utun1 (No description available)
6. utun2 (No description available)
7. utun3 (No description available)
8. utun4 (No description available)
9. utun5 (No description available)
10. utun7 (No description available)
11. utun8 (No description available)
12. lo0 (No description available)
13. anpi1 (No description available)
14. anpi0 (No description available)
15. en3 (No description available)
16. en4 (No description available)
17. en1 (No description available)
18. en2 (No description available)
19. bridge0 (No description available)
20. gif0 (No description available)
21. stf0 (No description available)
22. ap1 (No description available)
Seleccione el numero de interfez que desea: 1

Mensaje recibido por el usuario 1c:57:dc:74:a1:3f
Buenas tardes, esto es una prueba para la memoria de ReceiveFrames
```