INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Redes de Computadoras

“Analizador de tramas”

Versión 1.- LLC

Por:

- **Barenque Vázquez Diana Marisol**
- **Ramírez Benítez Brayan**

Profesora:
M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

Noviembre 2021

Introducción

Control de enlace lógico LLC (Logical Link Control) define la manera en que los datos son transferidos sobre un medio físico, concediendo servicio a las capas superiores. La cabecera LLC, incluye dos campos de dirección de 8 bits llamados DSAP (Destination Service Acces Point) la dirección LSAP que identifica la pila de protocolos en la computadora destino y SSAP (Source Service Acces Point) la dirección LSAP asociada a la pila de protocolos de origen en donde el bit menos significativo determina si es un comando o respuesta, un campo de control el cual contiene información que varía en función de la PDU y Datos recibidos de los protocolos de las capas superiores en forma de PDU de la capa de red.

Las tramas no numeradas (U) de 8 bits (*Figura 1)*, se utilizan para el gobierno del enlace en todo lo que se refiere a la conexión y desconexión, posee 5 bits M y un bit para el P/f, además, para M existen 12 comandos.

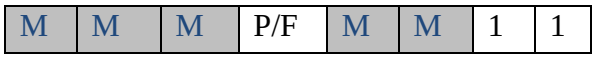


Figura 1. Estructura del campo de control T-U

Las tramas de información (I) de 16 bits (*Figura 2)*, se utilizan para transportar lo datos del usuario entre dos dispositivos de la red, el bit menos significativo contiene el p/f.

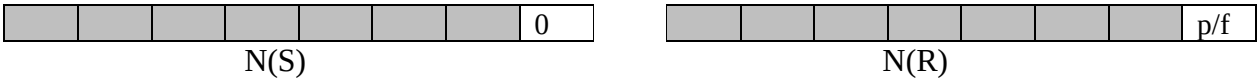


Figura 2. Estructura del campo de control T-I (16 bits)

Las tramas de supervisión (S) de 16 bits (*Figura 3)*, efectúan funciones de control, como bloqueos y desbloqueos de las transmisiones, la confirmación de las tramas recibidas, etc. El bit menos significativo contiene el p/f, con dos bits para S para el cual existen 4 comandos.

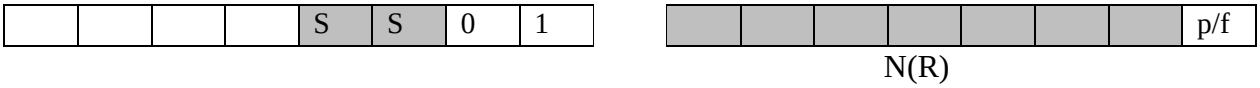


Figura 3. Estructura del campo de control T-S (16 bits)

P/F: (poll/final): bit sondeo/final. Solamente tiene significado cuando está activo y puede significar sondeo o final. Significa sondeo cuando la trama ha sido enviada desde una estación primaria a una secundaria y significa final cuando la trama se envía de una secundaria a un primario.

SAPo es el Punto de Acceso al servicio de origen indica si es un C/R (Comando/Respuesta). Si C/R es igual a 0 es comando, por el contrario, si C/R es igual a 1 es respuesta.

Explicación de los procesos de enmascaramiento

Para determinar el tipo de trama es necesario revisar el campo de control de nuestra cabecera LLC, pero para saber cuáles son los bytes del campo de control debemos consultar primero los 2 bytes del Tamaño/Tipo de la cabecera Ethernet. Al convertir los bytes de Tamaño/Tipo que están en hexadecimal a decimal, nos indicará el tamaño de bytes a considerar de la cabecera LLC. Los primeros dos bytes de la cabecera LLC se refieren al SAP Destino y Origen (como se indica en la siguiente imagen), después de estos bytes viene lo que nos interesa el campo de control, la *Figura 4* nos marca estos campos.

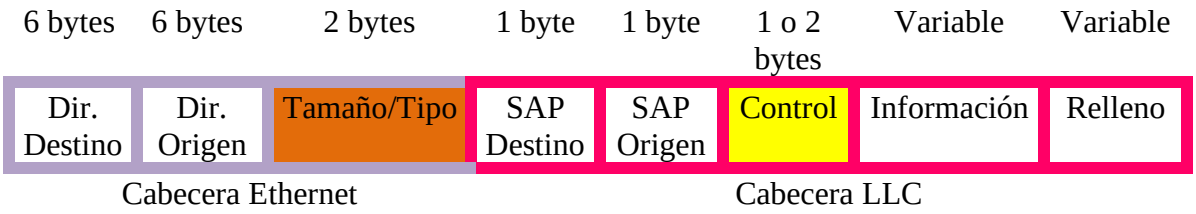


Figura 4. Estructura Trama Ethernet

El campo de control puede tener un tamaño de 1 byte o de 2 bytes como se muestra en las *Figura 5 y Figura 6*. Al convertir los bytes en hexadecimal a bits, revisaremos los primeros dos bits menos significativos del primer byte o único byte del campo de control. Si el bit menos significativo es 0 quiere decir que es una Trama de Información (T-I), pero sino revisaremos el segundo bit menos significativo. Si resulta ser 1, será una Trama No Numerada (T-U). En caso contrario es una Trama de Supervisión (T-S).

Figura 5. Campo de control de 2 bytes.

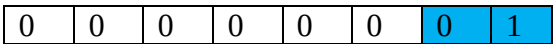


Figura 6. Campo de control de 1 byte

Para la obtención de los valores de los bits S de las Tramas de Supervisión que se pueden ver en la *Figura 3.* Revisamos el byte número 17 de nuestra trama, que es el primero byte del campo de control. En el código lo indicamos como T[16], le hicimos un corrimiento a la derecha de dos para quitar los primeros dos bits, como lo muestra la *Figura 7* (las X indican los valores de los bits que ya estaban).

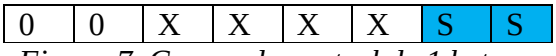


Figura 7. Campo de control de 1 byte

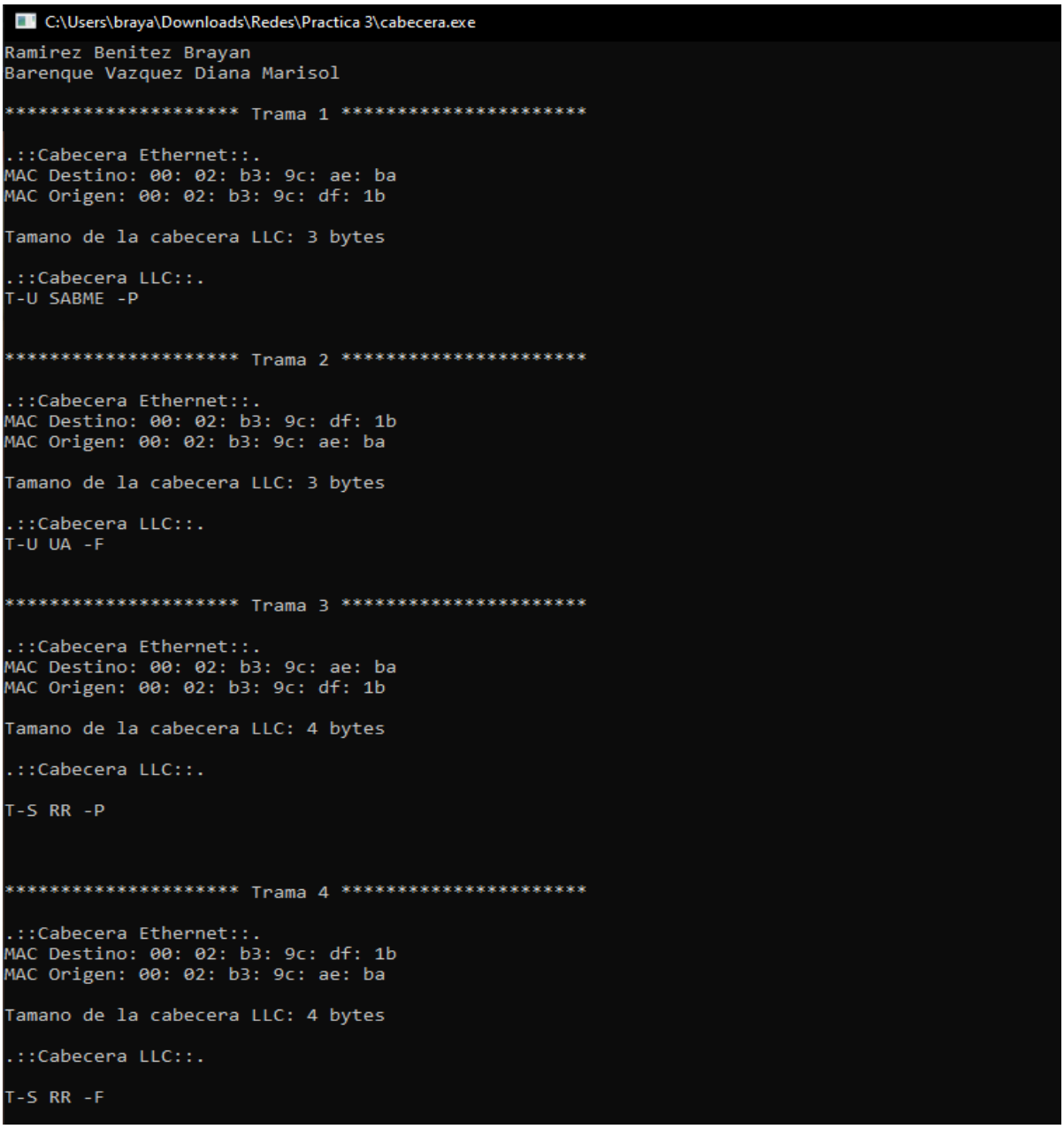
Luego hacemos un AND con un 3 (00000011), para así quedarnos solo con los valores de S. Dependiendo del valor que se obtenga de S se le asignará el comando.

$$\begin{array}{r} 00XXXXSS \\ 00000011 \\ \&\_\_\_\_\_ \\ 000000SS \end{array}$$

Para los bits M de las Tramas No Numeradas como se muestra en la *Figura 1*, tenemos que hacer un OR entre dos operaciones para juntar los bits M. Para la primera operación, de igual manera que con los bits S, se ocupara T[16] y también hacemos el corrimiento a la derecha de dos y hacemos un AND con 3. Para la segunda operación debemos hacer un corrimiento a la derecha de tres y un AND con 28 (00011100), para dejar espacio a los primeros dos bits M y con el OR concatenar los resultados de las dos operaciones. Así podemos conocer el valor del comando o respuesta.

$$\begin{array}{r} 00MMMXMM \\ 0\,0\,0\,0\,0\,0\,1\,1 \\ \&\_\_\_\_\_ \\ 0\,0\,0\,0\,0\,MM \end{array} \qquad \begin{array}{r} 000MMMXM \\ 000\,1\,1\,1\,0\,0 \\ \&\_\_\_\_\_ \\ 000MMM00 \end{array} \qquad \begin{array}{r} 000MMM00 \\ 000000MM \\ OR\_\_\_\_\_ \\ 000MMMMM \end{array}$$

Capturas de pantalla



Conclusiones

Ramirez Benitez Brayan

Antes de esta práctica ya había trabajado a nivel de bits, en la materia de estructura de datos para la elaboración de algunos programas. Generalmente no considero el uso de memoria puesto que me enfoco en resolver un problema y conseguir una buena complejidad temporal, sin embargo, la elaboración de esta practica y anteriores me han proporcionado nociones básicas para buscar optimizar el uso de memoria en los programas. Los operadores binarios nos permiten trabar en procesadores simples de bajo costo puesto que son más rápidos que los operadores que usamos normalmente como la suma, multiplicación o división. Durante la elaboración de la práctica necesitábamos un dominio muy completo sobre LLC para poder implementar la función que analiza las tramas, sin embargo, una vez que comprendemos el tema podemos realizar esta práctica de una manera relativamente sencilla.

Barenque Vázquez Diana Marisol

Lo más cercano a trabajar a nivel de bits a sido con las materias de Diseño de Sistemas Digital y Fundamentos de Diseño Digital al armar los circuitos y al programar, pero nunca lo había hecho en C o no lo recuerdo. Respecto al gasto de memoria, sinceramente no había comprendido la importancia de la memoria, se que esto son buenas prácticas, pero lo había escuchado mencionar de manera rápida. Por parte no había tenido la curiosidad y a veces es difícil entender, si alguien no te guía. Las ventajas que veo en los operadores binarios es el ahorro de inicialización de variables que solo te ayudaran a guardar por un rato algún resultado. Por ejemplo, cuando buscamos los bits M, no fue necesario crear una variable que guardara el resultado de la primera operación y otra para el segundo resultado, para después hacerles un OR, fue todo en una sola línea. También siento que después de comprender bien el funcionamiento de los operadores, es más fácil implementar así los programas y optimizar, al menos para este estilo de programas. La práctica si la hubiera hecho desde cero, sin apoyo de usted, si puedo decir que es difícil, pero le agradezco el tiempo y el detenimiento para explicarnos. No diré que fue fácil, porque los conocimientos adquiridos durante el semestre debes tenerlos muy claros y frescos, algunas cosas que se me fueron, pero ya repasé y me sentí lista para sacar este programa.

Código

```
#include<stdio.h>

void AnalizaTrama(unsigned char *);//Función tamaño o tipo
void AnalizaLLC(unsigned char *);

int main(){

        unsigned char i;
        unsigned char T[][192]={

        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x03,0xf0,0xf0,

0x7f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x43,0x05,0x9
0,0x6d}, //trama1


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x03,0xf0,0xf1,

0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x54,0x9
0,0x6d}, //trama2


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x04,0xf0,0xf0,

0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x41,0xa3,0x9
0,0x6d}, //trama3


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xf2,0x9
0,0x6d}, //trama4


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x12,0xf0,0xf0,

0x00,0x01,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7
f,0x23,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x41,0x9
1,0x6d}, //trama5


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x12,0xf0,0xf0,

0x00,0x03,0x0e,0x00,0xff,0xef,0x17,0x81,0xbc,0x05,0x23,0x00,0x7f,0x00,0x2
3,0x7f,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x90,0x9
1,0x6d}, //trama6


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,
```

```
0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xdf,0x9
1,0x6d}, //trama7


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x04,0xf0,0xf1,

0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0xac,0x9
2,0x6d}, //trama8


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0xac,0xf0,0xf0,

0x02,0x02,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7
f,0x23,

0xff,0x53,0x4d,0x42,0x72,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x8
2,0x09,

0x00,0x77,0x00,0x02,0x50,0x43,0x20,0x4e,0x45,0x54,0x57,0x4f,0x52,0x4b,0x2
0,0x50,

0x52,0x4f,0x47,0x52,0x41,0x4d,0x20,0x31,0x2e,0x30,0x00,0x02,0x4d,0x49,0x4
3,0x52,

0x4f,0x53,0x4f,0x46,0x54,0x20,0x4e,0x45,0x54,0x57,0x4f,0x52,0x4b,0x53,0x2
0,0x33,
```

```
0x2e,0x30,0x00,0x02,0x44,0x4f,0x53,0x20,0x4c,0x4d,0x31,0x2e,0x32,0x58,0x30,0x30,

0x32,0x00,0x02,0x44,0x4f,0x53,0x20,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x32,0x2e,0x31,

0x00,0x02,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x66,0x6f,0x72,0x20,0x57,0x6f,

0x72,0x6b,0x67,0x72,0x6f,0x75,0x70,0x73,0x20,0x33,0x2e,0x31,0x61,0x00,0x02,0x4e,

0x54,0x20,0x4c,0x4d,0x20,0x30,0x2e,0x31,0x32,0x00,0x00,0xfb,0x92,0x6d,0x86,0xdf}, //trama9


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,

0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7b,0x93,0x6d}, //trama10


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x5f,0xf0,0xf0,

0x02,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,

0xff,0x53,0x4d,0x42,0x72,0x00,0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,

0x11,0x05,0x00,0x02,0x02,0x00,0x01,0x00,0x68,0x0b,0x00,0x00,0x00,0x00,0x01,0x00,
```

```
0x7f,0x07,0x00,0x80,0x03,0x02,0x00,0x00,0x00,0xe5,0xfe,0x29,0x25,0x7c,0xc
2,0x01,

0x2c,0x01,0x08,0x08,0x00,0x7f,0x07,0x00,0x80,0x32,0x3e,0xb9,0x3d,0x00,0xc
a,0x93}, //trama11


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x04,0xf0,0xf1,

0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x9
4,0x6d}, //trama12


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x91,0xf0,0xf0,

0x04,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7
f,0x23,

0xff,0x53,0x4d,0x42,0x73,0x00,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x8
2,0x09,

0x0d,0x75,0x00,0x5d,0x00,0x68,0x0b,0x02,0x00,0x00,0x00,0x7f,0x07,0x00,0x8
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x20,0x00,0x00,0x0
0,0x45,

0x53,0x43,0x4f,0x4d,0x00,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x34,0x2
e,0x30,

0x00,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x34,0x2e,0x30,0x00,0x04,0xf
f,0x00,
```

```
0x00,0x00,0x02,0x00,0x02,0x00,0x17,0x00,0x20,0x00,0x5c,0x5c,0x50,0x52,0x4
f,0x47,

0x59,0x44,0x45,0x53,0x41,0x5c,0x49,0x50,0x43,0x24,0x00,0x49,0x50,0x43,0x0
0,0x00}, //trama13


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x32,0x9
5,0x6d}, //trama14


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x46,0xf0,0xf0,

0x04,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x2
3,0x7f,

0xff,0x53,0x4d,0x42,0x73,0x00,0x00,0x00,0x00,0x90,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x8
2,0x09,

0x03,0x75,0x00,0x29,0x00,0x00,0x00,0x00,0x00,0x02,0xff,0x00,0x00,0x00,0x0
4,0x00,

0x49,0x50,0x43,0x00,0x00,0x81,0x95,0x6d,0x86,0xcb,0x94,0x6d,0x86,0x0d,0x0
9,0x0e}, //trama15


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x04,0xf0,0xf1,

0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x9
6,0x6d}, //trama16


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x7e,0xf0,0xf0,

0x06,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7
f,0x23,

0xff,0x53,0x4d,0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x8
2,0x0a,

0x0e,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x88,0x1
3,0x00,

0x00,0x00,0x00,0x20,0x00,0x4c,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x0
0,0x5c,

0x50,0x49,0x50,0x45,0x5c,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x00,0x68,0x00,0x5
7,0x72,

0x4c,0x65,0x68,0x44,0x7a,0x00,0x42,0x31,0x36,0x42,0x42,0x44,0x7a,0x00,0x0
1,0x00,

0x00,0x10,0xff,0xff,0xff,0xff,0x45,0x53,0x43,0x4f,0x4d,0x00,0x00,0x6f,0x9
6,0x6d}, //trama17


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xbe,0x9
6,0x6d}, //trama18


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x04,0xf0,0xf1,

0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x5d,0x9
7,0x6d}, //trama19


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x7e,0xf0,0xf0,

0x08,0x08,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7
f,0x23,

0xff,0x53,0x4d,0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x0
2,0x0b,

0x0e,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x88,0x1
3,0x00,

0x00,0x00,0x00,0x20,0x00,0x4c,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x0
0,0x5c,

0x50,0x49,0x50,0x45,0x5c,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x00,0x68,0x00,0x5
7,0x72,

0x4c,0x65,0x68,0x44,0x7a,0x00,0x42,0x31,0x36,0x42,0x42,0x44,0x7a,0x00,0x0
1,0x00,

0x00,0x10,0x00,0x00,0x00,0x80,0x45,0x53,0x43,0x4f,0x4d,0x00,0x00,0xac,0x9
7,0x6d}, //trama20
```

```
        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xfb,0x9
7,0x6d}, //trama21


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x04,0xf0,0xf1,

0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x4a,0x9
8,0x6d}, //trama22


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x12,0xf0,0xf0,

0x0a,0x0b,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7
f,0x23,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x99,0x9
8,0x6d}, //trama23


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x0d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x45,0x9
9,0x6d}, //trama24


        {0x03,0x00,0x00,0x00,0x00,0x01,0x00,0x04,0xac,0x44,0x4d,0x02,0x00,
0x8b,0xf0,0xf0,

0x03,0x2c,0x00,0xff,0xef,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x42,0x3
4,0x20,

0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x1b,0x49,0x4
2,0x4d,

0x53,0x45,0x52,0x56,0x45,0x52,0x20,0x20,0x20,0x20,0x20,0x20,0x00,0xff,0x5
3,0x4d,

0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x11,0x0
0,0x00,

0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xe8,0x03,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x06,0x00,0x56,0x00,0x03,0x00,0x01,0x00,0x01,0x00,0x0
2,0x00,

0x17,0x00,0x5c,0x4d,0x41,0x49,0x4c,0x53,0x4c,0x4f,0x54,0x5c,0x42,0x52,0x4
f,0x57,

0x53,0x45,0x00,0x09,0x04,0x33,0x17,0x00,0x00,0x00,0x9b,0x99,0x6d,0x86,0x9
9,0x98}, //trama25


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x35,0xf0,0xf0,

0x0c,0x0a,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7
f,0x23,
```

```
0xff,0x53,0x4d,0x42,0x71,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x0
1,0x50,

0x00,0x00,0x00,0x45,0xf1,0x99,0x6d,0x86,0x45,0x99,0x6d,0x86,0x1f,0x09,0x5
2,0x5b}, //trama26


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x35,0xf0,0xf0,

0x0a,0x0e,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x2
3,0x7f,

0xff,0x53,0x4d,0x42,0x71,0x00,0x00,0x00,0x00,0x80,0x01,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x0
1,0x50,

0x00,0x00,0x00,0x00,0x40,0x9a,0x6d,0x86,0x9b,0x99,0x6d,0x86,0x20,0x09,0x7
5,0x5b}, //trama27


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x12,0xf0,0xf0,

0x0e,0x0d,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7
f,0x23,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x8f,0x9
a,0x6d}, //trama28


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xde,0x9
a,0x6d}, //trama29


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x12,0xf0,0xf0,

0x10,0x0d,0x0e,0x00,0xff,0xef,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7
f,0x23,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x9
b,0x6d}, //trama30


        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x04,0xf0,0xf1,

0x01,0x13,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x9
b,0x6d}, //trama31


        {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,
0x03,0xf0,0xf0,

0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xcb,0x9
b,0x6d}, //trama32
```

```
        {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,
0x03,0xf0,0xf1,

0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x77,0x9
c,0x6d},


        {0xff,0xff,0xff,0xff,0xff,0xff,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,
0x06,0x00,0x04,

0x08,0x00,0x06,0x04,0x00,0x01,0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x3
9,0xcb,
                0x00,0x00,0x00,0x00,0x00,0x00,0x94,0xcc,0x39,0xfe},
                                        //Trama a


        {0x00,0x23,0x8b,0x46,0xe9,0xad,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x08,
0x06,0x00,0x01,  //TRAMA b

0x08,0x00,0x06,0x04,0x00,0x02,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x94,0xcc,0x3
9,0xfe,

0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,0x00,0x00,0x00,0x00,0x0
0,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 },


        {0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,
0x00,0x46,0x00, // TRAMA c

0x80,0x42,0x04,0x55,0x34,0x11,0x80,0x11,0x6b,0xf0,0x94,0xcc,0x39,0xcb,0x9
4,0xcc,

0x67,0x02,0xaa,0xbb,0xcc,0xdd,0x04,0x0c,0x00,0x35,0x00,0x2e,0x85,0x7c,0xe
2,0x1a,

0x01,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x77,0x77,0x77,0x0
3,0x69,
```

```
0x73,0x63,0x05,0x65,0x73,0x63,0x6f,0x6d,0x03,0x69,0x70,0x6e,0x02,0x6d,0x7
8,0x00,
                0x00,0x1c,0x00,0x01}
                };

        printf("Ramirez Benitez Brayan\n");
        printf("Barenque Vazquez Diana Marisol\n\n");

        for(i = 0;i < 36;i++){
                printf("********************* Trama %d
**********************\n", i+1);
                AnalizaTrama(T[i]);
                puts("\n\n");
        }
        return 0;
}

void AnalizaTrama(unsigned char *T){

        printf("\n.::Cabecera Ethernet::.\n");
        printf("MAC Destino: %.2x: %.2x: %.2x: %.2x: %.2x: %.2x\n", T[0],
T[1], T[2], T[3], T[4], T[5]);
        printf("MAC Origen: %.2x: %.2x: %.2x: %.2x: %.2x: %.2x\n\n", T[6],
T[7], T[8], T[9], T[10], T[11]);

        unsigned short tot = (T[12]<<8)|T[13];

        if(tot < 1500){
                printf("Tamano de la cabecera LLC: %d bytes\n", tot);
                AnalizaLLC(T);
        }
        else{
                if(tot == 2048){
                        printf("Tipo IP\n");
                        //AnalizaIP(T);
                }else if(tot == 2054){
                        printf("Tipo ARP\n");
                        //AnalizaARP(T);
                }else{
                        printf("Otro %2.x %2.x\n", T[12], T[13]);
                }
        }

}
void AnalizaLLC(unsigned char *T){
```

```
        printf("\n.::Cabecera LLC::.\n");

        char ss[][5] = {"RR", "RNR", "REJ", "SREJ"};

        char uc[][6] = {"UI","SIM","-","SARM","UP","-","-
","SABM","DISC","-","-","SARME","-","-","-"
        ,"SABME","SNRM","-","-","RSET","-","-","-","XID","-","-","-
","SNRME","-","-","-","-"};

        char ur[][6] = {"UI","RIM","-","DM","-","-","-","-","RD","-","-
","-","UA","-","-"
        ,"-","-","FRMR","-","-","-","-","-","XID","-","-","-","-","-","-
","-","-"};

        switch(T[16]&3){
                case 0:
                        printf("\nT-I , N(s) = %d, N(r) = %d", T[16]>>1,
T[17]>>1);
                        if(T[17]&1){
                                if(T[15]&1)
                                        printf(" -F\n");
                                else
                                        printf(" -P\n");
                        }
                        break;
                case 1:
                        printf("\nT-S %s", ss[(T[16]>>2)&3]);
                        if(T[17]&1){
                                if(T[15]&1)
                                        printf(" -F\n");
                                else
                                        printf(" -P\n");
                        }
                        break;
                case 2:
                        printf("\nT-I , N(s) = %d, N(r) = %d", T[16]>>1,
T[17]>>1);
                        if(T[17]&1){
                                if(T[15]&1)
                                        printf(" -F\n");
                                else
                                        printf(" -P\n");
                        }
                        break;
                case 3:
                        if(T[16]&16){
```

```
                                if(T[15]&1)
                                        printf("T-U %s -F",
ur[((T[16]>>2)&3)|((T[16]>>3)&28)]);
                                else
                                        printf("T-U %s -P",
uc[((T[16]>>2)&3)|((T[16]>>3)&28)]);
                            }
                            break;
        }
}
```

Mapa de memoria