



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



## Redes de computadoras

### Práctica 3 “Mi trama”

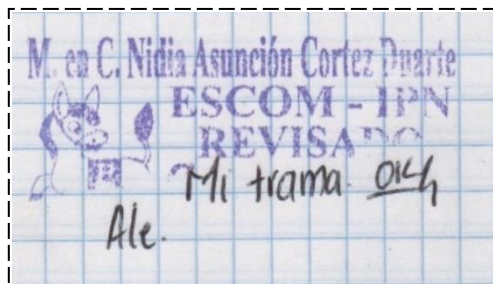
ALUMNO:

ZEPEDA FLORES ALEJANDRO DE JESÚS

PROFESOR:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

DICIEMBRE 2018



# Índice

Índice .....	2
Objetivo .....	3
Material y equipo .....	3
Marco teórico .....	3
Resultados .....	4
Conclusiones .....	6
Código.....	6
Bibliografía.....	9

## Objetivo

Desarrollar un programa en lenguaje C capaz de analizar una trama mediante el protocolo implementado, dicho programa recibirá una trama en forma de arreglo hexadecimal y analizará cada una de las 4 capas para imprimir las características del protocolo.

## Material y equipo

- Equipo de cómputo con el sistema operativos Windows
- Entorno de desarrollo (Sublime Text)
- Compilador para programas en C (GCC)

## Marco teórico

La trama es la unidad de datos que utiliza la capa de enlace. Más correctamente, según el estándar ISO/OSI, debería denominarse LPDU (Link Protocol Data Unit). Las tramas generadas por la capa de enlace de datos viajan desde un nodo hasta otro que está conectado al mismo medio físico de comunicación que el anterior. Una trama nunca salta de una subred a otra, sino que son el vehículo para transmitir datos que pueden estar viajando más allá de una subred, para atravesar un medio físico determinado.

La trama facilita la sincronización en la comunicación entre las entidades de la cada de Enlace, que consiste en la localización del comienzo y el final del bloque de información transmitido. Además, el protocolo ha de permitir la transmisión de cualquier tipo de datos que no deben de ser confundidos con información de control del protocolo, aunque su codificación coincida.

En una transmisión sincrónica se requiere de un nivel de sincronización adicional para que el receptor pueda determinar dónde está el comienzo y el final de cada bloque de datos. Normalmente, una trama constará de una cabecera, datos y cola. En la cola suele estar algún chequeo de errores. EN la cabecera habrá campos de control de protocolo y en la parte de datos es la que se quiere transmitir.

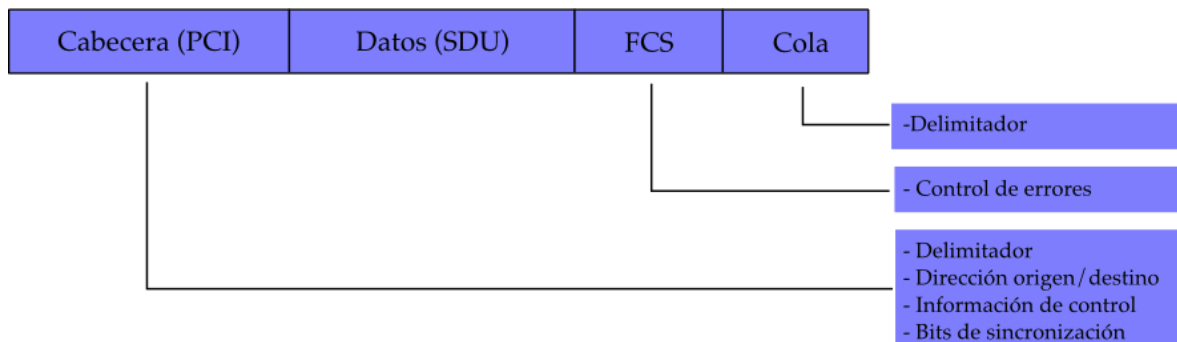


Imagen 1. Estructura básica de la trama

## Resultados

Para analizar los resultados, primero debemos mostrar la trama que se tiene que analizar. En el arreglo de caracteres trama, se encuentra declarada en formato hexadecimal.

```
int main(void){
    unsigned char cadena[100];
    unsigned char i = 0, aux = 0, size = 0, repeat = 0;
    unsigned char trama[] = {0x7B,0x69,0x6C,0x20,0x79,0x20,0x61,0x20,0x64,0x74,0x20,0x76,0x65,0x6E,0x74,0xCF,0x9A,0xA0,0x00,0x21};

    size = capa_uno(trama);
    size = capa_dos(trama,size);
    size = capa_tres(trama,size);
    size = capa_cuatro(trama,size);
    return 0;
}
```

Imagen 2. Declaración de la trama

Además de la imagen anterior, podemos rescatar que la trama es analizada capa por capa y de manera individual, evitando errores como de posicionamiento o análisis incorrecto dentro de la trama.

```
C:\Users\Alejandro ZF10\Desktop>gcc Práctica3.c -o P3
C:\Users\Alejandro ZF10\Desktop>P3

***** CAPA 1 *****

Size: 14 bytes
Language: French
Routing: Dinamic
Message: il y a dt vent

***** CAPA 2 *****

Grupo origen: 5CM3
Alumno origen: 15
Grupo destino: 3CM5
Alumno destino: 26

***** CAPA 3 *****

Control de flujo: Ventana deslizante
Tipo de control de flujo: GoBack N
Tipo de GoBack: 7 bits
Control de error: Bit de Paridad
Tipo de paridad: Paridad Par

***** CAPA 4 *****

Medio de transmision: Alámbrico
Tipo de medio de transmision: FO
Codigo de linea: RMI
C:\Users\Alejandro ZF10\Desktop>
```

Imagen 3. Impresión de protocolo

El análisis de la trama es relativamente, de manera sencilla, ya que la tarea fundamental del programa es analizar cada una de las posiciones de la misma, determinar en cual se encuentra y aplicar el análisis de la capa correspondiente según su posición.

Un dato relevante y fundamental dentro de nuestro análisis, es el tamaño de la trama (Ver imagen 2). El análisis se basa en esta variable, ya que es la limitante para nuestro ciclo que recorrerá la trama.

Sin este dato, no sería imposible, pero se dificultaría el análisis, ya que indica la pauta para el corrimiento.

0	0	0	0	0	0	0	0	Unsigned char size = 0;
0	1	1	1	1	0	1	1	Unsigned char trama[];
0	1	1	0	1	0	0	1	
0	1	1	0	1	1	0	0	
0	1	0	0	0	0	0	0	
0	1	1	1	1	0	0	1	
0	1	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	
0	1	0	0	0	0	0	0	
0	1	1	0	0	1	0	0	
0	1	1	1	0	1	0	0	
0	1	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	
0	1	1	0	0	1	0	1	
0	1	1	0	1	1	1	0	
0	1	1	1	0	1	0	0	
1	1	0	0	1	1	1	1	
1	0	0	1	1	0	1	0	
1	0	1	0	0	0	0	0	
0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	1	

Tabla 1. Mapa de memoria inicial

0	1	1	1	1	0	1	1	Unsigned char trama[];
0	1	1	0	1	0	0	1	
0	1	1	0	1	1	0	0	
0	1	0	0	0	0	0	0	
0	1	1	1	1	0	0	1	
0	1	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	
0	1	0	0	0	0	0	0	
0	1	1	0	0	1	0	0	
0	1	1	1	0	1	0	0	
0	1	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	
0	1	1	0	0	1	0	1	
0	1	1	0	1	1	1	0	
0	1	1	1	0	1	0	0	
1	1	0	0	1	1	1	1	
1	0	0	1	1	0	1	0	
1	0	1	0	0	0	0	0	
0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	1	

Tabla 2. Prueba de escritorio

## Conclusiones

El objetivo de la capa de enlace es lograr la comunicación entre 2 máquinas que estén conectadas fluya y sea libre. Para lograr este objetivo, se tienen que montar bloques de información, denominados tramas, para poder transmitir la información, gestionar o detectar la corrección de errores y ocuparse del control de flujo entre equipos. De ahí la importancia de la trama, para facilitar la transmisión de la información.

## Código

```
01. #include <stdio.h>
02. #include <stdlib.h>
03. #include <string.h>
04.
05. unsigned char bandera[6];
06.
07. int nobits(unsigned char cadena){
08.     unsigned char i = 0, j = 0, bits = 0;
09.     for(i=0; i<strlen(cadena) ;i++){
10.         for(j=128 ; j>0 ; j=j>>1){
11.             if(cadena[i]&j){
12.                 bits += 1;
13.                 printf("1");
14.             }
15.             else
16.                 printf("0");
17.         }
18.         printf(" ");
19.     }
20.     return bits;
21. }
22.
23. int getbits(unsigned char cadena){
24.     unsigned char i = 0, j = 0, bits = 0;
25.     for(j=128 ; j>0 ; j=j>>1){
26.         if(cadena&j)
27.             printf("1");
28.         else
29.             printf("0");
30.     }
31.     printf(" ");
32. }
33.
34. int capa_uno(unsigned char trama[]){
35.     unsigned char i = 0;
36.     unsigned char size = ((trama[0]>>4)*2);
37.     unsigned char enruta[][10] = {"Estatic", "Dinamic"};
38.     unsigned char idioma[][10] = {"-", "-", "Spanish", "English", "Portuguese", "French", "German"};
39.
40.     printf("\n***** CAPA 1 *****\n\n");
41.     printf("Size: %d bytes\n", size);
42.     printf("Language: %s\n", idioma[(trama[i]>>1)&7]);
43.     printf("Routing: %s\n", enruta[trama[i]&1]);
44.     printf("Message: ");
45.     for(i=1; i<(size+1) ;i++)
```

```

48.     return size+1;
49. }
50.
51. int capa_dos(unsigned char trama[], unsigned char size){
52.     unsigned char i = 0;
53.     unsigned char grupos[][5] = {"-", "2CM7", "3CM5", "5CM3"};
54.
55.     printf("\n\n***** CAPA 2 *****\n\n");
56.     printf("Grupo origen: %s\n", grupos[(trama[size]>>6)&3]);
57.     printf("Alumno origen: %d\n", trama[size++]&0x3F);
58.     printf("Grupo destino: %s\n", grupos[(trama[size]>>6)&3]);
59.     printf("Alumno destino: %d", trama[size++]&0x3F);
60.
61.     return size;
62. }
63.
64. int capa_tres(unsigned char trama[], unsigned char size){
65.     printf("\n\n***** CAPA 3 *****\n\n");
66.     switch((trama[size]>>7)&1){ //Control de flujo
67.         case 0: //Parar y esperar
68.             bandera[0] = 0;
69.             printf("Control de flujo: Parar y esperar\n");
70.             switch((trama[size]>>5)&3){
71.                 case 0:
72.                     bandera[3] = 0;
73.                     printf("Control de error: Bit de Paridad\n");
74.                     switch((trama[size]>>4)&1){
75.                         case 0: bandera[4] = 0; printf("Tipo de paridad: Paridad Par"); break;
76.                         case 1: bandera[4] = 1; printf("Tipo de paridad: Paridad Impar"); break;
77.                     }
78.                     break;
79.                 case 1:
80.                     bandera[3] = 1;
81.                     printf("Control de error: CRC\n");
82.                     switch((trama[size]>>4)&1){
83.                         case 0: bandera[5] = 0; printf("Tipo de CRC: 8 bits"); size = size + 1; break;
84.                         case 1: bandera[5] = 1; printf("Tipo de CRC: 32 bits"); size = size + 4; break;
85.                     }
86.                     break;
87.                 case 2:
88.                     bandera[3] = 2;
89.                     printf("Control de error: Checksum\n");

```

```

92.         break;
93.     case 1: //Ventana deslizante
94.         bandera[0] = 1;
95.         printf("Control de flujo: Ventana deslizante\n");
96.         switch((trama[size]>>6)&1){
97.             case 0: //GoBack N
98.                 bandera[1] = 0;
99.                 printf("Tipo de control de flujo: GoBack N\n");
100.                switch((trama[size]>>5)&1){
101.                    case 0: // 3 bits
102.                        bandera[2] = 0;
103.                        printf("Tipo de GoBack: 3 bits\n");
104.                        switch((trama[size]>>0)&3){
105.                            size = size + 1;
106.                            case 0:
107.                                bandera[3] = 0;
108.                                printf("Control de error: Bit de Paridad\n");
109.                                switch((trama[size]>>7)&1){
110.                                    case 0: bandera[4] = 0; printf("Tipo de paridad: Paridad Par"); break;
111.                                    case 1: bandera[4] = 1; printf("Tipo de paridad: Paridad Impar"); break;
112.                                }
113.                                break;
114.                                case 1:
115.                                    bandera[3] = 1;
116.                                    printf("Control de error: CRC\n");
117.                                    switch((trama[size]>>7)&1){
118.                                        case 0: bandera[5] = 0; printf("Tipo de CRC: 8 bits"); size = size + 1; break;
119.                                        case 1: bandera[5] = 1; printf("Tipo de CRC: 32 bits"); size = size + 4; break;
120.                                    }
121.                                    break;
122.                                    case 2:
123.                                        bandera[3] = 2;
124.                                        printf("Control de error: Checksum\n");
125.                                        break;
126.                                    }
127.                                break;
128.                                case 1: // 7 bits
129.                                    bandera[2] = 1;
130.                                    printf("Tipo de GoBack: 7 bits\n");
131.                                    size = size + 1;
132.                                    switch((trama[size]>>4)&3){
133.                                        case 0:
134.                                            bandera[3] = 0;
135.                                            printf("Control de error: Bit de Paridad\n");
136.                                            switch((trama[size]>>3)&1){
137.                                                case 0: bandera[4] = 0; printf("Tipo de paridad: Paridad Par"); break;
138.                                                case 1: bandera[4] = 1; printf("Tipo de paridad: Paridad Impar"); break;
139.                                            }
140.                                            break;
141.                                            case 1:
142.                                                bandera[3] = 1;
143.                                                printf("Control de error: CRC\n");
144.                                                switch((trama[size]>>3)&1){
145.                                                    case 0: bandera[5] = 0; printf("Tipo de CRC: 8 bits"); size = size + 1; break;
146.                                                    case 1: bandera[5] = 1; printf("Tipo de CRC: 32 bits"); size = size + 4; break;
147.                                                }
148.                                                break;
149.                                                case 2:
150.                                                    bandera[3] = 2;
151.                                                    printf("Control de error: Checksum\n");
152.                                                    break;
153.                                                }
154.                                            break;
155.                                        }
156.                                    break;
157.                                case 1: //Rechazo Selectivo
158.                                    bandera [1] = 0;
159.                                    printf("Tipo de control de flujo: Rechazo Selectivo\n");
160.                                    switch((trama[size]>>4)&3){
161.                                        case 0:
162.                                            bandera[3] = 0;
163.                                            printf("Control de error: Bit de Paridad\n");
164.                                            switch((trama[size]>>3)&1){
165.                                                case 0: bandera[4] = 0; printf("Tipo de paridad: Paridad Par"); break;
166.                                                case 1: bandera[4] = 1; printf("Tipo de paridad: Paridad Impar");break;
167.                                            }
168.                                            break;
169.                                            case 1:
170.                                                bandera[3] = 1;
171.                                                printf("Control de error: CRC\n");
172.                                                switch((trama[size]>>3)&1){
173.                                                    case 0: bandera[5] = 0; printf("Tipo de CRC: 8 bits"); size = size + 1; break;
174.                                                    case 1: bandera[5] = 1; printf("Tipo de CRC: 32 bits"); size = size + 4; break;
175.                                                }
176.                                                break;
177.                                                case 2:
178.                                                    bandera[3] = 2;
179.                                                    printf("Control de error: Checksum");
180.                                                    break;
181.                                                }
182.                                            break;

```



```

183.     }
184.     break;
185. }
186. return size;
187. }
188.
189. int capa_cuatro(unsigned char trama[], unsigned char size){
190.     unsigned char i = 0, j = 0;
191.     unsigned char codigo[][10] = {"Unipolar", "Polar", "RZ", "NRZ", "RMI"};
192.
193.     size = size + 1;
194.     printf("\n\n***** CAPA 4 *****\n\n");
195.     switch((trama[size]>>7)&1){
196.     case 0:
197.         printf("Medio de transmision: Alambrico\n");
198.         switch((trama[size]>>5)&3){
199.         case 1:
200.             printf("Tipo de medio de transmision: F0\n");
201.             break;
202.         case 2:
203.             printf("Tipo de medio de transmision: Coaxial\n");
204.             break;
205.         case 3:
206.             printf("Tipo de medio de transmision: UTP\n");
207.             break;
208.         }
209.         break;
210.     case 1:
211.         printf("Medio de transmision: inalambrico\n");
212.         switch((trama[size]>>5)&3){
213.         case 0:
214.             printf("Tipo de medio de transmision: Infrarrojo\n");
215.             break;
216.         case 2:
217.             printf("Tipo de medio de transmision: WiFi\n");
218.             break;
219.         case 3:
220.             printf("Tipo de medio de transmision: NFC\n");
221.             break;
222.         }
223.         break;
224.     }
225. }
226.
227. for(j=16; j>0; j=j>>1){
228.     if((trama[size]&31)&j)
229.         printf("Codigo de linea: %s\n",codigo[i++]);
230.     else
231.         i = i + 1;
232. }
233.
234. int main(void){
235.     unsigned char size = 0;
236.     unsigned char trama[] = {0x7B,0x69,0x6C,0x20,0x79,0x20,0x61,0x20,0x64,0x74,0x20,0x76,0x65,0x6E,0x74,0xCF,0x9A,0xA0,0x00,0x21};
237.
238.     size = capa_uno(trama);
239.     size = capa_dos(trama,size);
240.     size = capa_tres(trama,size);
241.     size = capa_cuatro(trama,size);
242.     printf("GABY %d",size);
243.     return 0;
244. }

```

## Bibliografía

- S/A. (2003). Capa de Enlace. Oviedo: Universidad de Oviedo.
- Sloman, M; Kramer, J. (1987) Distributed Systems and Computer Networks. Prentice-Hall