



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Redes de Computadoras

“Analizador de tramas”

Versión 1.- LLC

Por:

- **Martínez Coronel Brayan Yosafat**
- **Sánchez Méndez Edmundo Josué**

Profesora:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

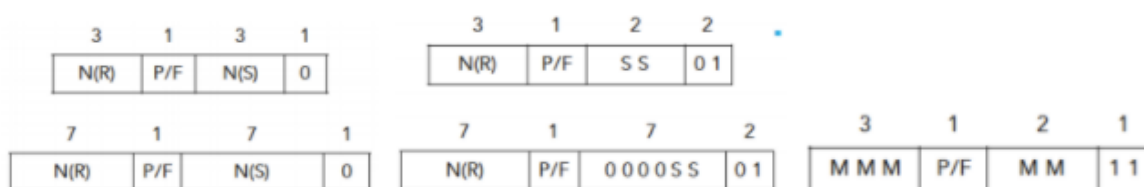
Enero 2021

PROTOCOLO LLC

El protocolo de Control del Enlace Lógico se usa en el enlace de las redes de área local, está descrito bajo el estándar IEEE 802.2, es bastante similar a las tramas de HDLC, esto es porque LLC está basado en HDLC, por lo que su formato es muy similar, está orientado a la conexión y los comandos y respuestas de las tramas no numeradas se mantienen al igual que las respuestas de las tramas supervisadas.

La cabecera LLC está compuesta de 4 partes, dirección de destino, dirección de origen, control LLC y el resto será la información, en ese orden respectivamente, ambas direcciones usan el bit menos significativo para simbolizar un valor booleano.

Como se mencionó, las tramas I, S y U se mantienen, y su estructura es similar a las originales en HDLC, la estructura de cada una se muestra respectivamente:



El bit P/F nos indica si es final o sondeo. Cuando es una trama de información, es sondeo cuando la manda una estación primaria a una secundaria, mientras si se envía al revés, será un final. N(R) y N(S) simbolizan el número de la trama que se transmite y la que se espera recibir, respectivamente. Siempre tienen 0 al final.

En la trama S, los bits S, pueden tomar 4 valores que en orden ascendente simbolizan RR, RNR, REJ y SREJ, sirven para proveer control del flujo de las tramas, por ejemplo, se manda RR cuando el Receptor está listo, o SREJ cuando se usará Rechazo Selectivo y siempre termina en 01. Mientras que los bits M en las tramas tienen algunas de las combinaciones y representan o bien un comando o respuesta según el bit P/F, estas establecen la conexión y siempre terminan en 11.

DETERMINACIÓN DEL TIPO DE TRAMA Y SUS BITS

Vamos a descomponer el problema, debemos tomar por hecho que es una cabecera LLC, o sea, ya pasó por esa verificación. Ahora, con este hecho, se sabe que el control LLC siempre estará fijo después de la dirección de origen, cuya posición es la 16 en el arreglo (contando el 0), al aplicar un and con 00000011, entonces sólo obtenemos los últimos dos bits que nos dan el tipo. El resto queda con los casos del 0 al 3, donde 0 y 2 simbolizan la trama de información, el 1 la de supervisión y el 3 la no numerada.

BITS S, M Y P/F

En la trama de información, tanto N(S) como N(R) tienen un bit que no representan las cuentas respectivamente. Con corrimiento a la derecha de 1 bit se obtienen ambas. Para la trama S el N(R) se obtiene de la misma forma, mientras que los bits S se obtienen con un corrimiento de 2 bits a la derecha de la posición 17 del arreglo. Se obtienen siempre que el bit P/F (que siempre está al final del primer byte del control LLC, ya que usamos el modo extendido).

Mientras que para la trama U, los bits M se tiene que usar un OR, desplazar la primera parte a la derecha por 3 bits, y la segunda parte a la derecha por 2 bits, con eso se obtienen los bits, pero, dependiendo del valor de la dirección de origen es que se usa un arreglo diferente para saber si es respuesta o comando.

CAPTURAS DE LA SALIDA

```
C:\Users\Edmundo J Sanchez M\Desktop>gcc AnalizadorLLC.c -o AnalizadorLLC
C:\Users\Edmundo J Sanchez M\Desktop>AnalizadorLLC.exe
Por:
    - Martinez Coronel Brayan Yosafat.
    - Sanchez Mendez Edmundo Josue.

TRAMA: 1
*****
      CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 3 bytes
      .::CABECERA LLC::.
T-U SABME P

*****

TRAMA: 2
*****
      CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 3 bytes
      .::CABECERA LLC::.
T-U UA F
```

TRAMA: 3

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=0, P
*****
```

TRAMA: 4

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=0, F
*****
```

TRAMA: 5

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 18 bytes
.:CABECERA LLC:..
T-I, N(s)= 0, N(r)=0, P
*****
```

TRAMA: 6

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 18 bytes
.:CABECERA LLC:..
T-I, N(s)= 0, N(r)=1, P
*****
```

TRAMA: 7

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=1, F
*****
```

TRAMA: 8

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=1, F
*****
```

TRAMA: 9

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 172 bytes
.:CABECERA LLC:..
T-I, N(s)= 1, N(r)=1,
*****
```

TRAMA: 10

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=2,
*****
```

TRAMA: 11

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 95 bytes
.:CABECERA LLC:..
T-I, N(s)= 1, N(r)=2,
*****
```

TRAMA: 12

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=2,
*****
```

TRAMA: 13

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 145 bytes
.:CABECERA LLC:..
T-I, N(s)= 2, N(r)=2,
*****
```

TRAMA: 14

```
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 4 bytes
.:CABECERA LLC:..
T-S, RR, N(r)=3,
*****
```

```

C:\Windows\system32\cmd.exe
*****
TRAMA: 15
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamano: 70 bytes
.:CABECERA LLC:.
T-I, N(s)= 2, N(r)=3,
*****
TRAMA: 16
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=3,
*****
TRAMA: 17
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 126 bytes
.:CABECERA LLC:.
T-I, N(s)= 3, N(r)=3,
*****
TRAMA: 18
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=4,
*****
TRAMA: 19
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=4,
*****

```

```

C:\Windows\system32\cmd.exe
TRAMA: 20
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 126 bytes
.:CABECERA LLC:.
T-I, N(s)= 4, N(r)=4,
*****
TRAMA: 21
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=5,
*****
TRAMA: 22
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=5,
*****
TRAMA: 23
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 18 bytes
.:CABECERA LLC:.
T-I, N(s)= 5, N(r)=5, P
*****
TRAMA: 24
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=6, F
*****

```

```

C:\Windows\system32\cmd.exe
TRAMA: 25
*****
CABECERA ETHERNET
MAC Destino: 03-00-00-00-00-01
MAC Origen: 00-04-ac-44-4d-02
Tamano: 139 bytes
.:CABECERA LLC:.
*****
TRAMA: 26
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 53 bytes
.:CABECERA LLC:.
T-I, N(s)= 6, N(r)=5,
*****
TRAMA: 27
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamano: 53 bytes
.:CABECERA LLC:.
T-I, N(s)= 5, N(r)=7,
*****
TRAMA: 28
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamano: 18 bytes
.:CABECERA LLC:.
T-I, N(s)= 7, N(r)=6, P
*****
TRAMA: 29
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamano: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=8, F
*****

```

```

TRAMA: 30
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 18 bytes
.:CABECERA LLC:.
T-I, N(s)= 8, N(r)=6, P
*****

TRAMA: 31
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 4 bytes
.:CABECERA LLC:.
T-S, RR, N(r)=9, F
*****

TRAMA: 32
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-ae-ba
MAC Origen: 00-02-b3-9c-df-1b
Tamaño: 3 bytes
.:CABECERA LLC:.
T-U DISC P

```

```

TRAMA: 33
*****
CABECERA ETHERNET
MAC Destino: 00-02-b3-9c-df-1b
MAC Origen: 00-02-b3-9c-ae-ba
Tamaño: 3 bytes
.:CABECERA LLC:.
T-U UA F
*****

TRAMA: 34
*****
CABECERA ETHERNET
MAC Destino: ff-ff-ff-ff-ff-ff
MAC Origen: 00-23-8b-46-e9-ad
TIPO ARP
*****

```

```

TRAMA: 35
*****
CABECERA ETHERNET
MAC Destino: 00-23-8b-46-e9-ad
MAC Origen: 00-1f-45-9d-1e-a2
TIPO ARP
*****

TRAMA: 36
*****
CABECERA ETHERNET
MAC Destino: 00-1f-45-9d-1e-a2
MAC Origen: 00-23-8b-46-e9-ad
TIPO IP
*****

C:\Users\Edmundo J Sanchez M\Desktop>

```

CÓDIGO

```

1. #include <stdio.h>
2. #define NumTramas 36
3.
4. unsigned char ss[][5] = {"RR", "RNR", "REJ", "SREJ"};
5. unsigned char uc[][6] = {"UI", "SIM", "-", "SARM", "UP", "-", "-", "SABM", "DISC", "-", "-", "SARME", "-", "-", "SABME", "SNRM", "-", "-", "RSET", "-", "-", "-", "XID", "-", "-", "-", "SNRME"};
6. unsigned char ur[][6] = {"UI", "RIM", "-", "DM", "-", "-", "-", "-", "RD", "-", "-", "-", "UA", "-", "-", "-", "FRMR", "-", "-", "-", "XID", "-", "-", "-", "-"};
7.
8. void analizarLLC(unsigned char *trama)
9. {
10.
11.     printf("\n\t.:CABECERA LLC:.\n");
12.     switch (trama[16] & 3)
13.     {
14.     case 0: // T-I
15.         printf("T-I, N(s)= %d, N(r)=%d, ", trama[16] >> 1, trama[17] >> 1);
16.         if (trama[17] & 1) //PF=1
17.         {
18.             if (trama[15] & 1) //LSB_SAPo=1
19.             {
20.                 printf("F\n");
21.             }
22.             else
23.             {
24.                 printf("P\n");
25.             }
26.         }

```

```

27.         break;
28.     case 1: // T-S
29.         printf("T-S, %s, N(r)=%d, ", ss[(trama[16] >> 2) & 3], trama[17] >> 1);
30.         if (trama[17] & 1) //PF=1
31.         {
32.             if (trama[15] & 1) //LSB_SAPo=1
33.             {
34.                 printf("F\n");
35.             }
36.             else
37.             {
38.                 printf("P\n");
39.             }
40.         }
41.         break;
42.     case 2: // T-I
43.         printf("T-I, N(s)= %d, N(r)=%d, ", trama[16] >> 1, trama[17] >> 1);
44.         if (trama[17] & 1) //PF=1
45.         {
46.             if (trama[15] & 1)
47.             {
48.                 printf("F\n");
49.             }
50.             else
51.             {
52.                 printf("P\n");
53.             }
54.         }
55.         break;
56.     case 3: // T-U
57.         if (trama[16] & 16) //PF=1
58.         {
59.             if (trama[15] & 1) //LSB_SAPo=1
60.             {
61.                 printf("T-
U %s F\n", ur[(((trama[16] >> 2) & 3) | ((trama[16] >> 3) & 28))]);
62.             }
63.             else
64.             {
65.                 printf("T-
U %s P\n", uc[(((trama[16] >> 2) & 3) | ((trama[16] >> 3) & 28))]);
66.             }
67.         }
68.         break;
69.     default:
70.         printf("%d", trama[16] & 3);
71.         break;
72.     }
73. }
74.
75. void analizartrama(unsigned char *trama)
76. {
77.     unsigned short int ToT = (trama[12] << 8) | trama[13];
78.     printf("\n*****\n\tCABECERA ETHERNET\n");
79.     printf("MAC Destino: %.2x-%.2x-%.2x-%.2x-%.2x-
%.2x\n", trama[0], trama[1], trama[2], trama[3], trama[4], trama[5]);
80.     printf("MAC Origen: %.2x-%.2x-%.2x-%.2x-%.2x-
%.2x\n", trama[6], trama[7], trama[8], trama[9], trama[10], trama[11]);
81.     if (ToT < 1500)
82.     {

```



```

124.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
125.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x41, 0x91, 0x6d}, //trama5
126.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x12, 0xf0, 0xf0,
127.      0x00, 0x03, 0x0e, 0x00, 0xff, 0xef, 0x17, 0x81, 0xbc, 0x05, 0
    x23, 0x00, 0x7f, 0x00, 0x23, 0x7f,
128.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
129.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x90, 0x91, 0x6d}, //trama6
130.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
131.      0x01, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
132.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
133.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0xdf, 0x91, 0x6d}, //trama7
134.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x04, 0xf0, 0xf1,
135.      0x01, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
136.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
137.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x18, 0xac, 0x92, 0x6d}, //trama8
138.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0xac, 0xf0, 0xf0,
139.      0x02, 0x02, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x04, 0x00, 0x00, 0
    x00, 0x00, 0x28, 0x00, 0x7f, 0x23,
140.      0xff, 0x53, 0x4d, 0x42, 0x72, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
141.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x82, 0x09,
142.      0x00, 0x77, 0x00, 0x02, 0x50, 0x43, 0x20, 0x4e, 0x45, 0x54, 0
    x57, 0x4f, 0x52, 0x4b, 0x20, 0x50,
143.      0x52, 0x4f, 0x47, 0x52, 0x41, 0x4d, 0x20, 0x31, 0x2e, 0x30, 0
    x00, 0x02, 0x4d, 0x49, 0x43, 0x52,
144.      0x4f, 0x53, 0x4f, 0x46, 0x54, 0x20, 0x4e, 0x45, 0x54, 0x57, 0
    x4f, 0x52, 0x4b, 0x53, 0x20, 0x33,
145.      0x2e, 0x30, 0x00, 0x02, 0x44, 0x4f, 0x53, 0x20, 0x4c, 0x4d, 0
    x31, 0x2e, 0x32, 0x58, 0x30, 0x30,
146.      0x32, 0x00, 0x02, 0x44, 0x4f, 0x53, 0x20, 0x4c, 0x41, 0x4e, 0
    x4d, 0x41, 0x4e, 0x32, 0x2e, 0x31,
147.      0x00, 0x02, 0x57, 0x69, 0x6e, 0x64, 0x6f, 0x77, 0x73, 0x20, 0
    x66, 0x6f, 0x72, 0x20, 0x57, 0x6f,
148.      0x72, 0x6b, 0x67, 0x72, 0x6f, 0x75, 0x70, 0x73, 0x20, 0x33, 0
    x2e, 0x31, 0x61, 0x00, 0x02, 0x4e,
149.      0x54, 0x20, 0x4c, 0x4d, 0x20, 0x30, 0x2e, 0x31, 0x32, 0x00, 0
    x00, 0xfb, 0x92, 0x6d, 0x86, 0xdf}, //trama9
150.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
151.      0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
152.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
153.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x7b, 0x93, 0x6d}, //trama10

```

```

154.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x5f, 0xf0, 0xf0,
155.      0x02, 0x04, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x0c, 0x00, 0x00, 0
    x28, 0x00, 0x28, 0x00, 0x23, 0x7f,
156.      0xff, 0x53, 0x4d, 0x42, 0x72, 0x00, 0x00, 0x00, 0x00, 0x80, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
157.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x82, 0x09,
158.      0x11, 0x05, 0x00, 0x02, 0x02, 0x00, 0x01, 0x00, 0x68, 0x0b, 0
    x00, 0x00, 0x00, 0x00, 0x01, 0x00,
159.      0x7f, 0x07, 0x00, 0x80, 0x03, 0x02, 0x00, 0x00, 0x00, 0x00, 0xe5, 0
    xfe, 0x29, 0x25, 0x7c, 0xc2, 0x01,
160.      0x2c, 0x01, 0x08, 0x08, 0x00, 0x7f, 0x07, 0x00, 0x80, 0x32, 0
    x3e, 0xb9, 0x3d, 0x00, 0xca, 0x93}, //trama11
161.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x04, 0xf0, 0xf1,
162.      0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
163.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
164.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x7c, 0x94, 0x6d}, //trama12
165.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x91, 0xf0, 0xf0,
166.      0x04, 0x04, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x0c, 0x00, 0x00, 0
    x28, 0x00, 0x28, 0x00, 0x7f, 0x23,
167.      0xff, 0x53, 0x4d, 0x42, 0x73, 0x00, 0x00, 0x00, 0x00, 0x10, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
168.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x82, 0x09,
169.      0x0d, 0x75, 0x00, 0x5d, 0x00, 0x68, 0x0b, 0x02, 0x00, 0x00, 0
    x00, 0x7f, 0x07, 0x00, 0x80, 0x00,
170.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0
    x00, 0x20, 0x00, 0x00, 0x00, 0x45,
171.      0x53, 0x43, 0x4f, 0x4d, 0x00, 0x57, 0x69, 0x6e, 0x64, 0x6f, 0
    x77, 0x73, 0x20, 0x34, 0x2e, 0x30,
172.      0x00, 0x57, 0x69, 0x6e, 0x64, 0x6f, 0x77, 0x73, 0x20, 0x34, 0
    x2e, 0x30, 0x00, 0xff, 0x00,
173.      0x00, 0x00, 0x02, 0x00, 0x02, 0x00, 0x17, 0x00, 0x20, 0x00, 0
    x5c, 0x5c, 0x50, 0x52, 0x4f, 0x47,
174.      0x59, 0x44, 0x45, 0x53, 0x41, 0x5c, 0x49, 0x50, 0x43, 0x24, 0
    x00, 0x49, 0x50, 0x43, 0x00, 0x00}, //trama13
175.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
176.      0x01, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00,
177.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
178.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x32, 0x95, 0x6d}, //trama14
179.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x46, 0xf0, 0xf0,
180.      0x04, 0x06, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x0c, 0x00, 0x00, 0
    x28, 0x00, 0x28, 0x00, 0x23, 0x7f,
181.      0xff, 0x53, 0x4d, 0x42, 0x73, 0x00, 0x00, 0x00, 0x00, 0x90, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
182.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0
    x00, 0x00, 0x00, 0x00, 0x82, 0x09,
183.      0x03, 0x75, 0x00, 0x29, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0
    xff, 0x00, 0x00, 0x00, 0x04, 0x00,

```

```

184.          0x49, 0x50, 0x43, 0x00, 0x00, 0x81, 0x95, 0x6d, 0x86, 0xcb, 0
x94, 0x6d, 0x86, 0x0d, 0x09, 0x0e}, //trama15
185.          {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
xdf, 0x1b, 0x00, 0x04, 0xf0, 0xf1,
186.          0x01, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
187.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
188.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x20, 0x96, 0x6d}, //trama16
189.          {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
xdf, 0x1b, 0x00, 0x7e, 0xf0, 0xf0,
190.          0x06, 0x06, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x0c, 0x00, 0x00, 0
x28, 0x00, 0x28, 0x00, 0x7f, 0x23,
191.          0xff, 0x53, 0x4d, 0x42, 0x25, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
192.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0
x00, 0x00, 0x00, 0x00, 0x82, 0x0a,
193.          0x0e, 0x20, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x10, 0x00, 0
x00, 0x00, 0x00, 0x88, 0x13, 0x00,
194.          0x00, 0x00, 0x00, 0x20, 0x00, 0x4c, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x2d, 0x00, 0x5c,
195.          0x50, 0x49, 0x50, 0x45, 0x5c, 0x4c, 0x41, 0x4e, 0x4d, 0x41, 0
x4e, 0x00, 0x68, 0x00, 0x57, 0x72,
196.          0x4c, 0x65, 0x68, 0x44, 0x7a, 0x00, 0x42, 0x31, 0x36, 0x42, 0
x42, 0x44, 0x7a, 0x00, 0x01, 0x00,
197.          0x00, 0x10, 0xff, 0xff, 0xff, 0xff, 0x45, 0x53, 0x43, 0x4f, 0
x4d, 0x00, 0x00, 0x6f, 0x96, 0x6d}, //trama17
198.          {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
199.          0x01, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
200.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
201.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0xbe, 0x96, 0x6d}, //trama18
202.          {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
xdf, 0x1b, 0x00, 0x04, 0xf0, 0xf1,
203.          0x01, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
204.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
205.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x5d, 0x97, 0x6d}, //trama19
206.          {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
xdf, 0x1b, 0x00, 0x7e, 0xf0, 0xf0,
207.          0x08, 0x08, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x0c, 0x00, 0x00, 0
x28, 0x00, 0x28, 0x00, 0x7f, 0x23,
208.          0xff, 0x53, 0x4d, 0x42, 0x25, 0x00, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x00, 0x00, 0x00,
209.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0
x00, 0x00, 0x00, 0x00, 0x02, 0x0b,
210.          0x0e, 0x20, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x10, 0x00, 0
x00, 0x00, 0x00, 0x88, 0x13, 0x00,
211.          0x00, 0x00, 0x00, 0x20, 0x00, 0x4c, 0x00, 0x00, 0x00, 0x00, 0
x00, 0x00, 0x00, 0x2d, 0x00, 0x5c,
212.          0x50, 0x49, 0x50, 0x45, 0x5c, 0x4c, 0x41, 0x4e, 0x4d, 0x41, 0
x4e, 0x00, 0x68, 0x00, 0x57, 0x72,
213.          0x4c, 0x65, 0x68, 0x44, 0x7a, 0x00, 0x42, 0x31, 0x36, 0x42, 0
x42, 0x44, 0x7a, 0x00, 0x01, 0x00,

```

```

214.      0x00, 0x10, 0x00, 0x00, 0x00, 0x80, 0x45, 0x53, 0x43, 0x4f, 0
    x4d, 0x00, 0x00, 0xac, 0x97, 0x6d}, //trama20
215.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
216.      0x01, 0x0a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
217.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
218.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0xfb, 0x97, 0x6d}, //trama21
219.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x04, 0xf0, 0xf1,
220.      0x01, 0x0a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
221.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
222.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x4a, 0x98, 0x6d}, //trama22
223.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x12, 0xf0, 0xf0,
224.      0x0a, 0x0b, 0x0e, 0x00, 0xff, 0xef, 0x14, 0x00, 0x00, 0x00, 0
    x28, 0x00, 0x00, 0x00, 0x7f, 0x23,
225.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
226.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x01, 0x99, 0x98, 0x6d}, //trama23
227.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
228.      0x01, 0x0d, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
229.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
230.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x45, 0x99, 0x6d}, //trama24
231.      {0x03, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x04, 0xac, 0x44, 0
    x4d, 0x02, 0x00, 0x8b, 0xf0, 0xf0,
232.      0x03, 0x2c, 0x00, 0xff, 0xef, 0x08, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x42, 0x34, 0x20,
233.      0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0
    x20, 0x20, 0x1b, 0x49, 0x42, 0x4d,
234.      0x53, 0x45, 0x52, 0x56, 0x45, 0x52, 0x20, 0x20, 0x20, 0x20, 0
    x20, 0x20, 0x00, 0xff, 0x53, 0x4d,
235.      0x42, 0x25, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
236.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x11, 0x00, 0x00,
237.      0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    xe8, 0x03, 0x00, 0x00, 0x00, 0x00,
238.      0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x56, 0x00, 0x03, 0x00, 0
    x01, 0x00, 0x01, 0x00, 0x02, 0x00,
239.      0x17, 0x00, 0x5c, 0x4d, 0x41, 0x49, 0x4c, 0x53, 0x4c, 0x4f, 0
    x54, 0x5c, 0x42, 0x52, 0x4f, 0x57,
240.      0x53, 0x45, 0x00, 0x09, 0x04, 0x33, 0x17, 0x00, 0x00, 0x00, 0
    x9b, 0x99, 0x6d, 0x86, 0x99, 0x98}, //trama25
241.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x35, 0xf0, 0xf0,
242.      0x0c, 0x0a, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x04, 0x00, 0x00, 0
    x00, 0x00, 0x28, 0x00, 0x7f, 0x23,
243.      0xff, 0x53, 0x4d, 0x42, 0x71, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x01, 0x00, 0x00, 0x00, 0x00, 0x00,

```

```

244.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0
    x00, 0x00, 0x00, 0x00, 0x01, 0x50,
245.      0x00, 0x00, 0x00, 0x45, 0xf1, 0x99, 0x6d, 0x86, 0x45, 0x99, 0
    x6d, 0x86, 0x1f, 0x09, 0x52, 0x5b}, //trama26
246.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x35, 0xf0, 0xf0,
247.      0x0a, 0x0e, 0x0e, 0x00, 0xff, 0xef, 0x16, 0x0c, 0x00, 0x00, 0
    x28, 0x00, 0x28, 0x00, 0x23, 0x7f,
248.      0xff, 0x53, 0x4d, 0x42, 0x71, 0x00, 0x00, 0x00, 0x00, 0x80, 0
    x01, 0x00, 0x00, 0x00, 0x00, 0x00,
249.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0
    x00, 0x00, 0x00, 0x00, 0x01, 0x50,
250.      0x00, 0x00, 0x00, 0x00, 0x40, 0x9a, 0x6d, 0x86, 0x9b, 0x99, 0
    x6d, 0x86, 0x20, 0x09, 0x75, 0x5b}, //trama27
251.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x12, 0xf0, 0xf0,
252.      0x0e, 0x0d, 0x0e, 0x00, 0xff, 0xef, 0x14, 0x00, 0x00, 0x00, 0
    x28, 0x00, 0x00, 0x00, 0x7f, 0x23,
253.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
254.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x8f, 0x9a, 0x6d}, //trama28
255.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
256.      0x01, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
257.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
258.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0xde, 0x9a, 0x6d}, //trama29
259.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x12, 0xf0, 0xf0,
260.      0x10, 0x0d, 0x0e, 0x00, 0xff, 0xef, 0x18, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x7f, 0x23,
261.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
262.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x2d, 0x9b, 0x6d}, //trama30
263.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x04, 0xf0, 0xf1,
264.      0x01, 0x13, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
265.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
266.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x7c, 0x9b, 0x6d}, //trama31
267.      {0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, 0xb3, 0x9c, 0
    xdf, 0x1b, 0x00, 0x03, 0xf0, 0xf0,
268.      0x53, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
269.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
270.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0xcb, 0x9b, 0x6d}, //trama32
271.      {0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, 0xb3, 0x9c, 0
    xae, 0xba, 0x00, 0x03, 0xf0, 0xf1,
272.      0x73, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
273.      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

```

274.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00, 0x00, 0x77, 0x9c, 0x6d}, //trama 33
275.
276.          {0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, 0x8b, 0x46, 0
    xe9, 0xad, 0x08, 0x06, 0x00, 0x04,
277.          0x08, 0x00, 0x06, 0x04, 0x00, 0x01, 0x00, 0x23, 0x8b, 0x46, 0
    xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb,
278.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, 0x39, 0xfe},
    /*Trama a */
279.
280.          {0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, 0x45, 0x9d, 0
    x1e, 0xa2, 0x08, 0x06, 0x00, 0x01, /*TRAMA b */
281.          0x08, 0x00, 0x06, 0x04, 0x00, 0x02, 0x00, 0x1f, 0x45, 0x9d, 0
    x1e, 0xa2, 0x94, 0xcc, 0x39, 0xfe,
282.          0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, 0
    x00, 0x00, 0x00, 0x00, 0x00, 0x00,
283.          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x00, 0x00},
284.
285.          {0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, 0x8b, 0x46, 0
    xe9, 0xad, 0x08, 0x06, 0x46, 0x00, /* TRAMA c */
286.          0x80, 0x42, 0x04, 0x55, 0x34, 0x11, 0x80, 0x11, 0x6b, 0xf0, 0
    x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc,
287.          0x67, 0x02, 0xaa, 0xbb, 0xcc, 0xdd, 0x04, 0x0c, 0x00, 0x35, 0
    x00, 0x2e, 0x85, 0x7c, 0xe2, 0x1a,
288.          0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
    x03, 0x77, 0x77, 0x77, 0x03, 0x69,
289.          0x73, 0x63, 0x05, 0x65, 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0
    x70, 0x6e, 0x02, 0x6d, 0x78, 0x00,
290.          0x00, 0x1c, 0x00, 0x01}};
291.
292.    printf("Por: \n");
293.    printf("\t- Martinez Coronel Brayan Yosafat.\n");
294.    printf("\t- Sanchez Mendez Edmundo Josue.\n");
295.    for (i; i < NumTramas; i++)
296.    {
297.        printf("\nTRAMA: %i\n", i+1);
298.        analizartrama(trama[i]);
299.    }
300. }

```

CONCLUSIONES

MARTÍNEZ CORONEL BRAYAN YOSAFAT

Aunque sí me preocupo de las optimizaciones y del uso de memoria, nunca había sido al grado en el que lo hacemos aquí. Es mi primera vez trabajando al nivel de bits y veo la gran utilidad de C, gastan lo necesario para operaciones que en sí gastarían mucho más si se usan los otros métodos por desconocer cómo funcionan esos operadores. Esta práctica no fue complicada, pero eso es porque nos hizo practicar antes con los operadores, lo que ahora veo con otro enfoque, me hubiera costado mucho más si hubiera seguido pensando a un nivel alto de programación.

Es la primera vez en el tiempo que llevo en ESCOM sin contar las materias de sistemas digitales en donde trabajo a nivel de bits y es que generalmente no me preocupo de la cantidad de memoria que gasto, si no que solo me enfoco con que funcione correctamente, además, vemos como trabajar a nivel de bits y en especial usando los operadores binarios hacemos más optimo nuestro código ya que las comparaciones son más rápidas y en general la practica fue medianamente difícil ya que al no estar acostumbrado a programar a nivel de bits si es un poco confuso al inicio pero una vez que entendí como se trabaja a nivel de bits fue más fácil acabarla.