|  |  |  |
| --- | --- | --- |
| **ipn** | **INSTITUTO POLITÉCNICO NACIONAL**  **ESCUELA SUPERIOR DE CÓMPUTO** |  |

**Redes de Computadoras**

**“ARP Address Resolution Protocol”**

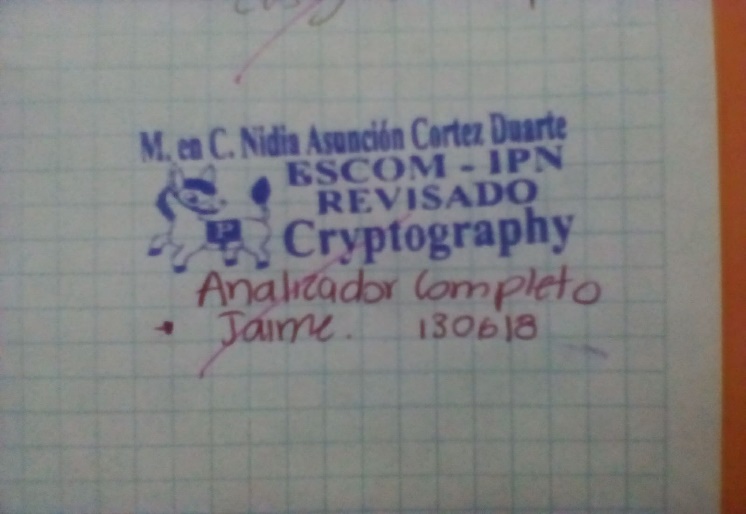
**By:**

**Bastida Prado Jaime Armando**

Professor:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

May 2018

****

**Index**

Contenido

[Problem: 1](#_Toc381082154)

[Hypothesis: 2](#_Toc381082155)

[Software (libraries, packages, tools): 2](#_Toc381082156)

[Procedure: 2](#_Toc381082157)

[Results (Data): 2](#_Toc381082158)

[Conclusions: 5](#_Toc381082159)

[References: 5](#_Toc381082160)

[Code 5](#_Toc381082161)

# Problem:

In this practice we are going to continue with the program for analyzing LLC frames by adding support for ARP frames.

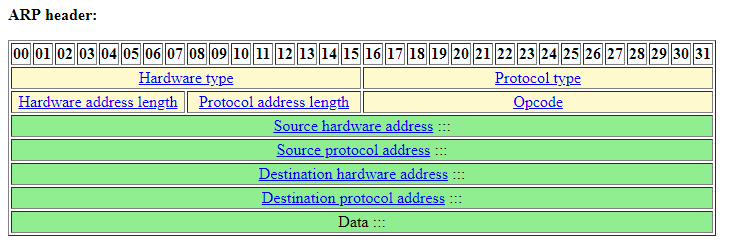
ARP is a network layer address resolution protocol. It is used to translate protocol addresses to hardware interface addresses and vice versa.

The ARP header goes after the MAC header and before the data:



*Figure 1: Ethernet header*

The ARP header looks like this:



*Figure 2: ARP header*

The **EtherType** for ARP corresponds to the value **0x0806 = 2054**.

**Hardware types** are usually:

* 00 01 Ethernet
* 00 06 Token Ring
* 00 0F Frame Relay
* 00 10 Atm

The only **Protocol type** existing at the moment:

* 08 00 IP

**Hardware** and **Protocol address length** are just that the lengths in bytes.

There are several **Operation Codes** but the most used are:

* 1 = ARP Request
* 2 = ARP Replay
* 8 = Inverse ARP Request
* 9 = Inverse ARP Replay

**Source hardware** and **protocol addresses** are variable in length depending on the values given before. **Destination hardware** and **protocol addresses** are variable too.

# Hypothesis:

The solution to the problem is to develop a program to analyze ARP frames relaying on the theory given by the teacher in class.

# Software (libraries, packages, tools):

The equipment used:

-Laptop

-Pencil

-Sheets

-The program was codified in C language using text editor Sublime Text 3

-Compiler: GNU (GCC) compiler.

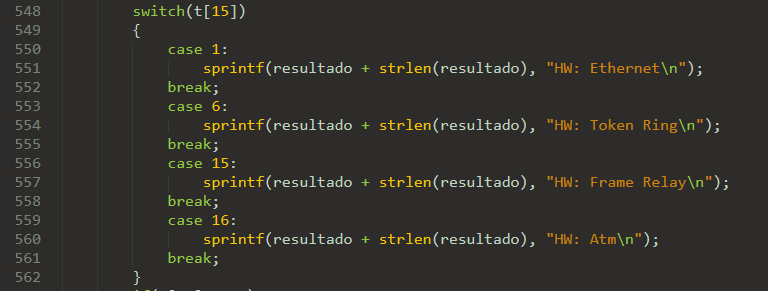
# Procedure:

Follow the steps given in the LLC report up to the point where the ToT variable is checked for the value 2054 inside an if statement



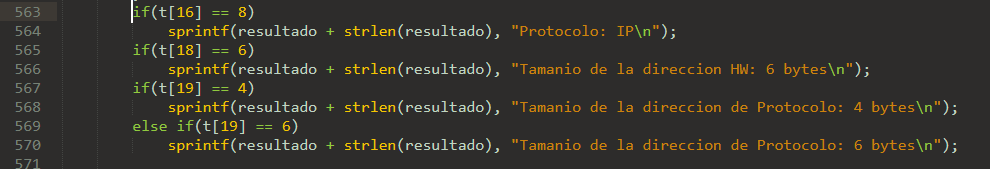
*Figure 3: ARP case*

Now, we need to check the **HW** **type**, to do so just write a switch statement as follows



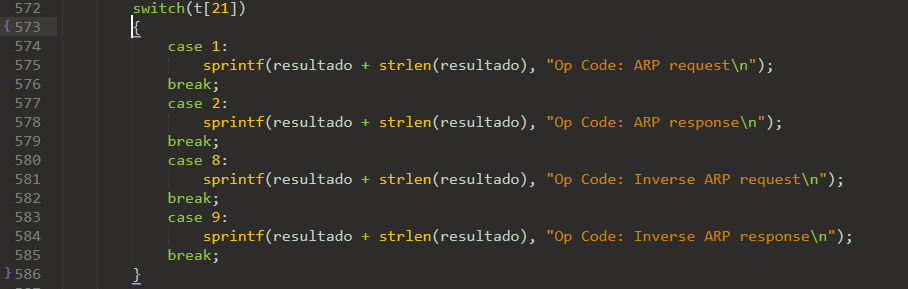
*Figure 4: Hardware types switch*

Check the **Protocol type** (although there is only IP) and the lengths of the addresses



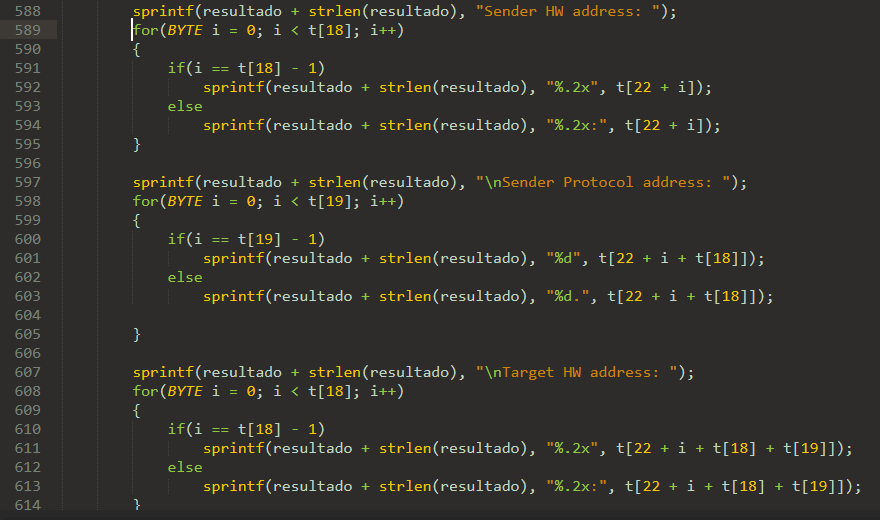
*Figure 5: Printing Protocol type check and lengths of the addresses*

Then, check the **Op Code** (we are only going to check for the four most commonly used)



*Figure 6: Op Code switch*

Finally, print the correspondent addresses



*Figure 7: Printing the addresses*

You may notice that I put this particular if statement inside each for

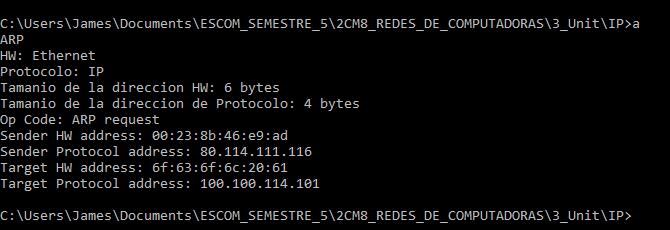


*Figure 8: Particular if statement in for*

Its purpose is to avoid printing an address like this: “192.34.56.123.” with a dot at the end.

# Results (Data):

Testing the program with an ARP frame given by the teacher we can observe the results of our program:



*Figure 9: Result ARP shown in console*

# Conclusions:

In conclusion our program does the analysis as requested by the teacher with no problems, it is using the minimum amount of data storage as required for spare. Now I understand better how all the work done below the table is when using the Internet.

# References:

K.N. King, C Programming A Modern Approach Second Edition. W.W. Norton , 2008.

Network Sorcery. (2012). ARP, Address Resolution Protocol. Retrieved from <http://www.networksorcery.com/enp/protocol/arp.htm>

Code

1. #include <stdio.h>
2. #include <string.h>
3. #include <stdlib.h>
5. **typedef** unsigned **char** **BYTE**;
7. **int** analizaTrama(**BYTE** t[], **BYTE** resultado[]);
9. **int** main(**void**)
10. {
11. //LLC frames
12. /\*BYTE t[ ][32] = { {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0, 0x7f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 1
14. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1, 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 2
16. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf0, 0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//Trama 3
18. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 4
20. {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,0x00,0x23,0x00,0x7f,0x23}, //Trama 5
22. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x12,0xf0,0xf0, 0x00,0x03,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23}, //Trama 6
24. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 7
26. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 8
28. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0xac,0xf0,0xf0, 0x02,0x02,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 9
30. {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}, //Trama 10
32. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x5f,0xf0,0xf0, 0x02,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 11
34. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 12
36. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x91,0xf0,0xf0, 0x04,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 13
38. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 14
40. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x46,0xf0,0xf0, 0x04,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 15
42. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 16
44. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0, 0x06,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 17
46. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 18
48. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 19
50. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0, 0x08,0x08,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 20
52. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 21
54. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x0a,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23}, //Trama 22
56. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x0a,0x0b,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23}, //Trama 23
58. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x0d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 24
60. {0x03,0x00,0x00,0x00,0x00,0x01,0x00,0x04,0xac,0x44,0x4d,0x02,0x00,0x8b,0xf0,0xf0, 0x03,0x2c,0x00,0xff,0xef,0x08,0x00,0x00,0x00,0x00,0x00,0x28,0x00,0x42,0x34,0x20}, //Trama 25
62. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0, 0x0c,0x0a,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 26
64. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0, 0x0a,0x0e,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 27
66. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x0e,0x0d,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23}, //Trama 28
68. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 29
70. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x10,0x0d,0x0e,0x00,0xff,0xef,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7f,0x23}, //Trama 30
72. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x13,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 31
74. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0, 0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 32
76. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1, 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}}; //Trama 33\*/
77. //ARP frames
78. //Remove the "t1" I made that way because I wanted to prove something else
79. **BYTE** t1[20][125]={
80. /\* TRAMA 1 \*/
81. {
82. 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, /\* .......# \*/
83. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x04, /\* .F...... \*/
84. 0x08, 0x00, 0x06, 0x04, 0x00, 0x01, 0x00, 0x23, /\* .......# \*/
85. 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, /\* .F....9. \*/
86. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, /\* ........ \*/
87. 0x39, 0xfe                                      /\* 9. \*/
88. },
90. /\*TRAMA 2 \*/
91. {
92. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
93. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x06, 0x00, 0x01, /\* E....... \*/
94. 0x08, 0x00, 0x06, 0x04, 0x00, 0x02, 0x00, 0x1f, /\* ........ \*/
95. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x39, 0xfe, /\* E.....9. \*/
96. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
97. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
98. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
99. 0x00, 0x00, 0x00, 0x00                          /\* .... \*/
100. },
102. /\* TRAMA 3 \*/
103. {
104. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
105. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x46, 0x00, /\* .F....F. \*/
106. 0x80, 0x42, 0x04, 0x55, 0x34, 0x11, 0x80, 0x11, /\* .B.U4... \*/
107. 0x6b, 0xf0, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* k...9... \*/
108. 0x67, 0x02, 0xaa, 0xbb, 0xcc, 0xdd, 0x04, 0x0c, /\* g....... \*/
109. 0x00, 0x35, 0x00, 0x2e, 0x85, 0x7c, 0xe2, 0x1a, /\* .5...|.. \*/
110. 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
111. 0x00, 0x00, 0x03, 0x77, 0x77, 0x77, 0x03, 0x69, /\* ...www.i \*/
112. 0x73, 0x63, 0x05, 0x65, 0x73, 0x63, 0x6f, 0x6d, /\* sc.escom \*/
113. 0x03, 0x69, 0x70, 0x6e, 0x02, 0x6d, 0x78, 0x00, /\* .ipn.mx. \*/
114. 0x00, 0x1c, 0x00, 0x01                          /\* .... \*/
115. },
117. /\* TRAMA 4 \*/
118. {
119. 0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, /\* ........ \*/
120. 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x04, 0xf0, 0xf0, /\* ........ \*/
121. 0x01, 0xc9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
122. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
123. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
124. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
125. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
126. 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x9b, 0x6d  /\* .....|.m \*/
127. },
129. /\* TRAMA 5 \*/
130. {
131. 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, /\* .......# \*/
132. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x10, /\* .F...... \*/
133. 0x08, 0x00, 0x06, 0x04, 0x00, 0x03, 0x00, 0x23, /\* .......# \*/
134. 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, /\* .F....9. \*/
135. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, /\* ........ \*/
136. 0x3a, 0xe1                                      /\* :. \*/
137. },
139. /\* TRAMA 6\*/
140. {
141. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
142. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x06, 0x00, 0x10, /\* E....... \*/
143. 0x08, 0x00, 0x06, 0x04, 0x00, 0x04, 0x00, 0x1f, /\* ........ \*/
144. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x3a, 0xe1, /\* E.....:. \*/
145. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
146. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
147. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
148. 0x00, 0x00, 0x00, 0x00                          /\* .... \*/
149. },
151. /\* TRAMA 7 \*/
152. {
153. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
154. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x00, 0x45, 0x00, /\* E.....E. \*/
155. 0x00, 0x6f, 0x90, 0x30, 0x40, 0x00, 0xfb, 0x11, /\* .o.0@... \*/
156. 0x24, 0xe7, 0x94, 0xcc, 0x67, 0x02, 0x94, 0xcc, /\* $...g... \*/
157. 0x39, 0xcb, 0x00, 0x35, 0x04, 0x0c, 0x00, 0x5b, /\* 9..5...[ \*/
158. 0xe8, 0x60, 0xe2, 0x1a, 0x85, 0x80, 0x00, 0x01, /\* .`...... \*/
159. 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x03, 0x77, /\* .......w \*/
160. 0x77, 0x77, 0x03, 0x69, 0x73, 0x63, 0x05, 0x65, /\* ww.isc.e \*/
161. 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0x70, 0x6e, /\* scom.ipn \*/
162. 0x02, 0x6d, 0x78, 0x00, 0x00, 0x1c, 0x00, 0x01, /\* .mx..... \*/
163. 0xc0, 0x14, 0x00, 0x06, 0x00, 0x01, 0x00, 0x00, /\* ........ \*/
164. 0x0e, 0x10, 0x00, 0x21, 0x04, 0x64, 0x6e, 0x73, /\* ...!.dns \*/
165. 0x31, 0xc0, 0x1a, 0x03, 0x74, 0x69, 0x63, 0xc0, /\* 1...tic. \*/
166. 0x1a, 0x77, 0xec, 0xdf, 0x29, 0x00, 0x00, 0x2a, /\* .w..)..\* \*/
167. 0x30, 0x00, 0x00, 0x0e, 0x10, 0x00, 0x12, 0x75, /\* 0......u \*/
168. 0x00, 0x00, 0x00, 0x2a, 0x30                    /\* ...\*0 \*/
169. },
171. /\* TRAMA 8 \*/
172. {
173. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
174. 0x02, 0x03, 0x04, 0x05, 0x08, 0x00, 0x45, 0x00, /\* .F....E. \*/
175. 0x00, 0x42, 0x04, 0x56, 0x00, 0x00, 0x80, 0x11, /\* .B.V.... \*/
176. 0x6b, 0xef, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* k...9... \*/
177. 0x67, 0x02, 0x04, 0x0c, 0x00, 0x35, 0x00, 0x2e, /\* g....5.. \*/
178. 0xff, 0x87, 0x68, 0x2a, 0x01, 0x00, 0x00, 0x01, /\* ..h\*.... \*/
179. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x77, /\* .......w \*/
180. 0x77, 0x77, 0x03, 0x69, 0x73, 0x63, 0x05, 0x65, /\* ww.isc.e \*/
181. 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0x70, 0x6e, /\* scom.ipn \*/
182. 0x02, 0x6d, 0x78, 0x00, 0x00, 0x01, 0x00, 0x01  /\* .mx..... \*/
183. },
185. /\* TRAMA 9 \*/
186. {
187. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
188. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x45, 0x00, /\* .F....E. \*/
189. 0x00, 0x3c, 0x04, 0x57, 0x00, 0x00, 0x80, 0x01, /\* .<.W.... \*/
190. 0x98, 0x25, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* .%..9... \*/
191. 0x3a, 0xe1, 0x08, 0x00, 0x49, 0x5c, 0x03, 0x00, /\* :...I\.. \*/
192. 0x01, 0x00, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, /\* ..abcdef \*/
193. 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, /\* ghijklmn \*/
194. 0x6f, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, /\* opqrstuv \*/
195. 0x77, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, /\* wabcdefg \*/
196. 0x68, 0x69                                      /\* hi \*/
197. },
199. /\* TRAMA 10 \*/
200. {
201. 0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, /\* ........ \*/
202. 0x02, 0x03, 0x04, 0x05, 0x00, 0x04, 0xf0, 0xf0, /\* ........ \*/
203. 0x0c, 0xc7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
204. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
205. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
206. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
207. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
208. 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x9b, 0x6d  /\* .....|.m \*/
209. },
211. /\* trama 11 \*/
212. {
213. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x11, 0x11, /\* ........ \*/
214. 0x11, 0x11, 0x11, 0x11, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
215. 0x63, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
216. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
217. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
218. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
219. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
220. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
221. },
223. /\* trama 12 \*/
224. {
225. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
226. 0x02, 0x03, 0x04, 0x05, 0x08, 0x00, 0x45, 0x00, /\* E.....E. \*/
227. 0x00, 0x3c, 0x01, 0xb5, 0x00, 0x00, 0x3f, 0x01, /\* .<....?. \*/
228. 0xdb, 0xc7, 0x94, 0xcc, 0x3a, 0xe1, 0x94, 0xcc, /\* ....:... \*/
229. 0x39, 0xcb, 0x00, 0x00, 0x51, 0x5c, 0x03, 0x00, /\* 9...Q\.. \*/
230. 0x01, 0x00, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, /\* ..abcdef \*/
231. 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, /\* ghijklmn \*/
232. 0x6f, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, /\* opqrstuv \*/
233. 0x77, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, /\* wabcdefg \*/
234. 0x68, 0x69                                      /\* hi \*/
235. },
237. /\* trama 13 \*/
238. {
239. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x01, /\* ........ \*/
240. 0x02, 0x03, 0x04, 0x05, 0x00, 0x03, 0xf0, 0xf1, /\* ........ \*/
241. 0x73, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
242. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
243. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
244. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
245. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
246. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
247. },
249. /\* trama 14 \*/
250. {
251. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x1c, /\* ..E..... \*/
252. 0xc0, 0x7b, 0x35, 0xa1, 0x08, 0x00, 0x48, 0x00, /\* .{5...H. \*/
253. 0x00, 0x48, 0x5c, 0x7d, 0x00, 0x00, 0x80, 0x01, /\* .H\}.... \*/
254. 0x6c, 0x88, 0x94, 0xcc, 0x39, 0xc3, 0x94, 0xcc, /\* l...9... \*/
255. 0x00, 0x49, 0x07, 0x0b, 0x04, 0x00, 0x00, 0x00, /\* .I...... \*/
256. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, /\* ........ \*/
257. 0x3b, 0x5c, 0x02, 0x00, 0x10, 0x00, 0x61, 0x62, /\* ;\....ab \*/
258. 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69, 0x6a, /\* cdefghij \*/
259. 0x6b, 0x6c, 0x6d, 0x6e, 0x6f, 0x70, 0x71, 0x72, /\* klmnopqr \*/
260. 0x73, 0x74, 0x75, 0x76, 0x77, 0x61, 0x62, 0x63, /\* stuvwabc \*/
261. 0x64, 0x65, 0x66, 0x67, 0x68, 0x69              /\* defghi \*/
262. },
264. /\* trama 15\*/
265. {
266. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
267. 0x45, 0x9d, 0x1e, 0xa2, 0x80, 0x35, 0x00, 0x01, /\* E....5.. \*/
268. 0x08, 0x00, 0x06, 0x04, 0x00, 0x03, 0x00, 0x1f, /\* ........ \*/
269. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x3a, 0xe1, /\* E.....:. \*/
270. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
271. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
272. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
273. 0x00, 0x00, 0x00, 0x00, 0xd8, 0xee, 0xdf, 0xb0  /\* ........ \*/
274. },
276. /\* trama 16 \*/
277. {
278. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x22, 0x22, /\* ........ \*/
279. 0x22, 0x22, 0x22, 0x22, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
280. 0x73, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
281. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
282. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
283. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
284. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
285. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
286. },
288. /\*trama 17 \*/
289. {
290. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, /\* ........ \*/
291. 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
292. 0x53, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
293. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
294. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
295. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
296. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
297. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
298. },
300. /\* trama 18 \*/
301. {
302. 0x02, 0xff, 0x53, 0xc3, 0xe9, 0xab, 0x00, 0xff, /\* ..S..... \*/
303. 0x66, 0x7f, 0xd4, 0x3c, 0x08, 0x00, 0x45, 0x00, /\* f..<..E. \*/
304. 0x00, 0x30, 0x2c, 0x00, 0x40, 0x00, 0x80, 0x06, /\* .0,.@... \*/
305. 0x4b, 0x74, 0xc0, 0xa8, 0x01, 0x02, 0xc0, 0xa8, /\* Kt...... \*/
306. 0x01, 0x01, 0x04, 0x03, 0x00, 0x15, 0x00, 0x3b, /\* .......; \*/
307. 0xcf, 0x44, 0x00, 0x00, 0x00, 0x00, 0x70, 0x20, /\* .D....p  \*/
308. 0x20, 0x00, 0x0c, 0x34, 0x00, 0x00, 0x02, 0x04, /\*  ..4.... \*/
309. 0x05, 0xb4, 0x01, 0x01, 0x04, 0x02              /\* ...... \*/
310. },
312. /\*trama 19 \*/
313. {
314. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, /\* ........ \*/
315. 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x03, 0xf0, 0xf1, /\* ........ \*/
316. 0x53, 0x0b, 0x0e, 0x00, 0xff, 0xef, 0x14, 0x00, /\* ........ \*/
317. 0x00, 0x00, 0x28, 0x00, 0x00, 0x00, 0x7f, 0x23, /\* ..(....# \*/
318. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
319. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
320. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
321. 0x00, 0x00, 0x00, 0x00, 0x01, 0x99, 0x98, 0x6d  /\* .......m \*/
322. },
323. /\* trama 20 \*/
324. {
325. 0x00, 0xff, 0x66, 0x7f, 0xd4, 0x3c, 0x02, 0xff, /\* ..f..<.. \*/
326. 0x53, 0xc3, 0xe9, 0xab, 0x08, 0x00, 0x45, 0x00, /\* S.....E. \*/
327. 0x00, 0x30, 0x05, 0xc4, 0x40, 0x00, 0x80, 0x06, /\* .0..@... \*/
328. 0x71, 0xb0, 0xc0, 0xa8, 0x01, 0x01, 0xc0, 0xa8, /\* q....... \*/
329. 0x01, 0x02, 0x00, 0x15, 0x04, 0x03, 0x21, 0x5d, /\* ......!] \*/
330. 0x3a, 0x44, 0x00, 0x3b, 0xcf, 0x45, 0x70, 0x12, /\* :D.;.Ep. \*/
331. 0x44, 0x70, 0x8c, 0x11, 0x00, 0x00, 0x02, 0x04, /\* Dp...... \*/
332. 0x05, 0xb4, 0x01, 0x01, 0x04, 0x02              /\* ...... \*/
333. }};
335. **BYTE** t[] = {0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x01,
336. 0x08, 0x00, 0x06, 0x04, 0x00, 0x01, 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb,
337. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, 0x39, 0xfe};
338. //For IP
339. /\*
340. BYTE t[] = {0x13, 0x42, 0x18, 0x63, 0x96, 0x12, 0x13, 0x42,  0x18, 0x96, 0x35, 0x72, 0x08, 0x00, 0x46, 0x00,
341. 0x80, 0x42, 0x04, 0x55, 0x34, 0x11, 0x80, 0x11,  0x6b, 0xf0, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc,
342. 0x67, 0x02, 0xaa, 0xbb, 0xcc, 0xdd, 0x04, 0x0c,  0x00, 0x35, 0x00, 0x2e, 0x85, 0x7c, 0xe2, 0x1a,
343. 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x03, 0x77, 0x77, 0x77, 0x03, 0x69,
344. 0x73, 0x63, 0x05, 0x65, 0x73, 0x63, 0x6f, 0x6d,  0x03, 0x69, 0x70, 0x6e, 0x02, 0x6d, 0x78, 0x00,
345. 0x00, 0x1c, 0x00, 0x01};
346. \*/
347. **BYTE** resultado[150];
349. //For just one frame
351. **if**(analizaTrama(t, resultado))
352. printf("%s\n", resultado);
353. **else**
354. printf("No es trama LLC\n");

357. //For LLC frames
358. /\*
359. for(BYTE i = 0; i < 33; i++)
360. if(analizaTrama(t[i], resultado))
361. printf("%d %s\n", i + 1, resultado);
362. else
363. printf("No es trama LLC\n");
364. \*/
366. //For ARP frames
367. /\*
368. for(BYTE i = 0; i < 5; i++)
369. if(analizaTrama(t[i], resultado))
370. printf("%d %s\n\n", i + 1, resultado);
371. else
372. printf("No es trama LLC\n");
373. \*/
374. exit(EXIT\_SUCCESS);
375. }
377. **int** analizaTrama(**BYTE** t[], **BYTE** resultado[])
378. {
379. **BYTE** supervision[][5] = {"RR", "RNR", "REJ", "SREJ"};
380. **BYTE** uc[][6] = {"UI", "SIM", "-", "SARN", "UP", "-", "-", "SABM", "DISC", "-", "-", "SARME", "-", "-", "-", "SABME",
381. "SNRM", "-", "-", "XID", "-", "-", "-", "-", "-", "-", "-", "SNRME"};
382. **BYTE** ur[][5] = {"UI", "RIM", "-", "DM", "-", "-", "-", "-", "RD", "-", "-", "-", "UA", "-", "-", "-", "-", "FRMR", "-",
383. "XID", "-", "-", "-", "-", "-", "-", "-", "-"};
384. unsigned **short** **int** ToT = 0;
386. //Copiamos los bytes 12 y 13 de la trama en ToT (Tamaño o tipo)
387. **for**(**BYTE** j = 128; j > 0; j >>= 1)
388. **if**(t[12] & j)
389. ToT = (ToT << 1) | 1;
390. **else**
391. ToT <<= 1;
393. **for**(**BYTE** j = 128; j > 0; j >>= 1)
394. **if**(t[13] & j)
395. ToT = (ToT << 1) | 1;
396. **else**
397. ToT <<= 1;
399. //Tipo LLC
400. **if**(ToT <= 1500)
401. {
402. **switch**(t[16] & 3)
403. {
404. **case** 0: **case** 2:
405. sprintf(resultado, "T-I N(s) = %d, N(r) = %d", ((t[16] >> 1) & 255), ((t[17] >> 1) & 255));
406. **break**;
407. **case** 1:
408. sprintf(resultado, "T-S %s, N(r) = %d", supervision[(t[16] >> 2) & 3], ((t[17] >> 1) & 255));
409. **break**;
410. **case** 3:
411. **if**(((t[16] >> 4) & 1) && (t[15] & 1)) //Respuesta no numerada
412. sprintf(resultado, "T-U %s", ur[((t[16] >> 2) & 3) | ((t[16] >> 3) & 28)]);
413. **else** **if**(((t[16] >> 4) & 1) && !(t[15] & 0)) //Comando no numerado
414. sprintf(resultado, "T-U %s", uc[((t[16] >> 2) & 3) | ((t[16] >> 3) & 28)]);
415. **else** //?
416. sprintf(resultado, "Trama Intrusa");
417. **break**;
418. }
420. **return** 1;
421. }
422. **else** **if**(ToT == 2048) //Tipo IP
423. {
424. sprintf(resultado, "IP\n");
425. sprintf(resultado + strlen(resultado), "Version: %d\n", (t[14] >> 4));
426. sprintf(resultado + strlen(resultado), "IHL: %d bytes\n", (t[14] & 15) \* 4);
427. sprintf(resultado + strlen(resultado), "Tipo Servico: ");
428. **if**(t[15] & 2)
429. sprintf(resultado + strlen(resultado), "| Maximize Delay | ");
430. **if**(t[15] & 4)
431. sprintf(resultado + strlen(resultado), "| Maximize Throughout | ");
432. **if**(t[15] & 8)
433. sprintf(resultado + strlen(resultado), "| Maximize Reliability | ");
434. **if**(t[15] & 16)
435. sprintf(resultado + strlen(resultado), "| Minimize Cost |");
436. sprintf(resultado + strlen(resultado), "\n");
438. sprintf(resultado + strlen(resultado), "Tamanio total: %d\n", (t[16] << 8) + t[17]);
439. sprintf(resultado + strlen(resultado), "Identificador: %d\n", (t[18] << 8) + t[19]);
440. sprintf(resultado + strlen(resultado), "Offset: %d saltos en bytes", (((t[20] & 31) << 8) + t[21]) \* 8);
441. **if**(t[20] & 64)
442. sprintf(resultado + strlen(resultado), "| Don't fragment ");
443. **else**
444. sprintf(resultado + strlen(resultado), "| Fragmented ");
445. **if**(t[20] & 64)
446. sprintf(resultado + strlen(resultado), "| More fragments ");
447. **else**
448. sprintf(resultado + strlen(resultado), "| No more fragments ");
449. sprintf(resultado + strlen(resultado), "\n");
451. sprintf(resultado + strlen(resultado), "TTL: %d\n", t[22]);
452. sprintf(resultado + strlen(resultado), "Protocolo: ");
453. **if**(t[23] & 1)
454. sprintf(resultado + strlen(resultado), "ICMP\n");
455. **else** **if**(t[23] & 6)
456. sprintf(resultado + strlen(resultado), "TCP\n");
457. **else** **if**(t[23] & 17)
458. sprintf(resultado + strlen(resultado), "UDP\n");
459. **else**
460. sprintf(resultado + strlen(resultado), "Otro\n");
462. sprintf(resultado + strlen(resultado), "Checksum Trama: %4x\n", (t[24] << 8) + t[25]);
463. unsigned **short** checksum = 0;
464. unsigned **short** temp = 0;
466. **for**(**int** j = 128; j > 0; j >>= 1)
467. **if**(t[14] & j)
468. checksum = (checksum << 1) | 1;
469. **else**
470. checksum <<= 1;
471. **for**(**int** j = 128; j > 0; j >>= 1)
472. **if**(t[15] & j)
473. checksum = (checksum << 1) | 1;
474. **else**
475. checksum <<= 1;
477. //sprintf(resultado + strlen(resultado), "Checksum: %4x\n", checksum);
478. //No calculamos los t[24] ni t[25] porque corresponden al checksum previamente calculado
479. **for**(**int** i = 16; i < 14 + ((t[14] & 15) \* 4); i++)
480. {
481. **if**(i != 24)
482. {
483. **for**(**int** j = 128; j > 0; j >>= 1)
484. **if**(t[i] & j)
485. temp = (temp << 1) | 1;
486. **else**
487. temp <<= 1;
488. i++;
489. **for**(**int** j = 128; j > 0; j >>= 1)
490. **if**(t[i] & j)
491. temp = (temp << 1) | 1;
492. **else**
493. temp <<= 1;
495. **if**(((checksum >> 12) + (temp >> 12)) > 15)
496. checksum += 1;
497. //sprintf(resultado + strlen(resultado), "Temp: %4x\n", temp);
498. checksum += temp;
499. //sprintf(resultado + strlen(resultado), "Checksum: %4x\n", checksum);
500. //sprintf(resultado + strlen(resultado), "\n");
501. }
502. **else**
503. i++;
504. }
505. checksum = ~checksum;
506. sprintf(resultado + strlen(resultado), "Checksum Calculado: %4x\n", checksum);
507. **if**(checksum != ((t[24] << 8) + t[25]))
508. sprintf(resultado + strlen(resultado), "Error! :(\n");
509. **else**
510. sprintf(resultado + strlen(resultado), "Correcto! :)\n");
512. sprintf(resultado + strlen(resultado), "IP Origen: ");
513. **for**(**int** i = 26; i < 26 + (t[14] >> 4); i++)
514. {
515. **if**(i == (25 + (t[14] >> 4)))
516. sprintf(resultado + strlen(resultado), "%d", t[i]);
517. **else**
518. sprintf(resultado + strlen(resultado), "%d.", t[i]);
519. }
520. sprintf(resultado + strlen(resultado), "\nIP Destino: ");
521. **for**(**int** i = 30; i < 30 + (t[14] >> 4); i++)
522. {
523. **if**(i == (29 + (t[14] >> 4)))
524. sprintf(resultado + strlen(resultado), "%d", t[i]);
525. **else**
526. sprintf(resultado + strlen(resultado), "%d.", t[i]);
527. }
528. **if**(((t[14] & 15) \* 4) == 20)
529. sprintf(resultado + strlen(resultado), "\nNo hay opciones");
530. **else**
531. {
532. sprintf(resultado + strlen(resultado), "\nOpciones: ");
533. **for**(**int** i = 34; i < 34 + (((t[14] & 15) \* 4) - 20); i++)
534. {
535. **if**(i == 33 + (((t[14] & 15) \* 4) - 20))
536. sprintf(resultado + strlen(resultado), "%2x", t[i]);
537. **else**
538. sprintf(resultado + strlen(resultado), "%2x:", t[i]);
539. }
540. }
542. **return** 1;
543. }
544. **else** **if**(ToT == 2054) //ARP
545. {
546. sprintf(resultado, "ARP\n");
547. **switch**(t[15])
548. {
549. **case** 1:
550. sprintf(resultado + strlen(resultado), "HW: Ethernet\n");
551. **break**;
552. **case** 6:
553. sprintf(resultado + strlen(resultado), "HW: Token Ring\n");
554. **break**;
555. **case** 15:
556. sprintf(resultado + strlen(resultado), "HW: Frame Relay\n");
557. **break**;
558. **case** 16:
559. sprintf(resultado + strlen(resultado), "HW: Atm\n");
560. **break**;
561. }
562. **if**(t[16] == 8)
563. sprintf(resultado + strlen(resultado), "Protocolo: IP\n");
564. **if**(t[18] == 6)
565. sprintf(resultado + strlen(resultado), "Tamanio de la direccion HW: 6 bytes\n");
566. **if**(t[19] == 4)
567. sprintf(resultado + strlen(resultado), "Tamanio de la direccion de Protocolo: 4 bytes\n");
568. **else** **if**(t[19] == 6)
569. sprintf(resultado + strlen(resultado), "Tamanio de la direccion de Protocolo: 6 bytes\n");
571. **switch**(t[21])
572. {
573. **case** 1:
574. sprintf(resultado + strlen(resultado), "Op Code: ARP request\n");
575. **break**;
576. **case** 2:
577. sprintf(resultado + strlen(resultado), "Op Code: ARP response\n");
578. **break**;
579. **case** 8:
580. sprintf(resultado + strlen(resultado), "Op Code: Inverse ARP request\n");
581. **break**;
582. **case** 9:
583. sprintf(resultado + strlen(resultado), "Op Code: Inverse ARP response\n");
584. **break**;
585. }
587. sprintf(resultado + strlen(resultado), "Sender HW address: ");
588. **for**(**BYTE** i = 0; i < 6; i++)
589. {
590. **if**(i == 5)
591. sprintf(resultado + strlen(resultado), "%.2x", t[22 + i]);
592. **else**
593. sprintf(resultado + strlen(resultado), "%.2x:", t[22 + i]);
594. }
595. sprintf(resultado + strlen(resultado), "\nSender Protocol address: ");
596. **for**(**BYTE** i = 0; i < 4; i++)
597. {
598. **if**(i == 3)
599. sprintf(resultado + strlen(resultado), "%d", t[22 + i + 6]);
600. **else**
601. sprintf(resultado + strlen(resultado), "%d.", t[22 + i + 6]);
603. }
604. sprintf(resultado + strlen(resultado), "\nTarget HW address: ");
605. **for**(**BYTE** i = 0; i < 6; i++)
606. {
607. **if**(i == 5)
608. sprintf(resultado + strlen(resultado), "%.2x", t[22 + i + 6 + 4]);
609. **else**
610. sprintf(resultado + strlen(resultado), "%.2x:", t[22 + i + 6 + 4]);
611. }
612. sprintf(resultado + strlen(resultado), "\nTarget Protocol address: ");
613. **for**(**BYTE** i = 0; i < 4; i++)
614. {
615. **if**(i == 3)
616. sprintf(resultado + strlen(resultado), "%d", t[22 + i + 6 + 4 + 6]);
617. **else**
618. sprintf(resultado + strlen(resultado), "%d.", t[22 + i + 6 + 4 + 6]);
619. }
621. **return** 1;
622. }
624. **return** 0;
625. }