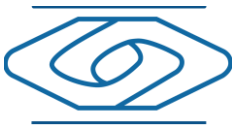
Instituto Politécnico Nacional.

Escuela Superior de Cómputo.



Materia: Redes de Computadoras.

Tema: Análisis de Tramas Ethernet.

Profesor: Axel Ernesto Moreno Cervantes.

Integrantes:

Rojas Alvarado Luis Enrique.

Miranda Sandoval Mario Alberto.

Grupo: 2CM10.

Introducción

El control de enlace lógico o LLC (Logical link control) por sus siglas en inglés pertenece a la familia de estándares IEEE 802 para el control de enlace de datos. LLC utiliza algunas características de HDLC además de que incluye algunas que no tiene el anterior. En LLC las funciones para controlar el enlace se dividen en dos capas la capa de control de acceso al medio y la capa LLC que funciona por encima de la capa MAC.

(c) LLC/MAC

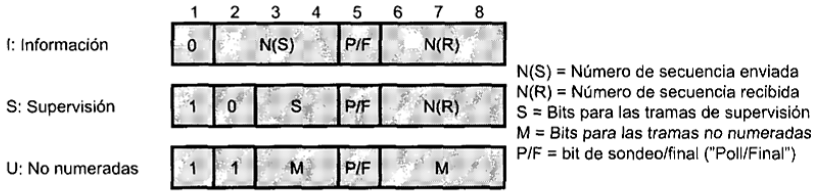
La MAC incluye las direcciones origen y destino para identificar a los dispositivos conectados en la LAN. Estas dos direcciones son necesarias ya que en el entorno LAN no existe el concepto de estación primaria o secundaria por lo que el emisor y el receptor deben ser identificados. La detección de errores se realiza en el nivel MAC utilizando CRC de 32 bits.

En la capa LLC hay cuatro campos. Los puntos de acceso al servicio del destino y del origen los cuales identifican al usuario lógico del LLC en los sistemas origen y destino. El campo de control del LLC tiene el campo de control limitado a la utilización de números de secuencia de 7 bits.

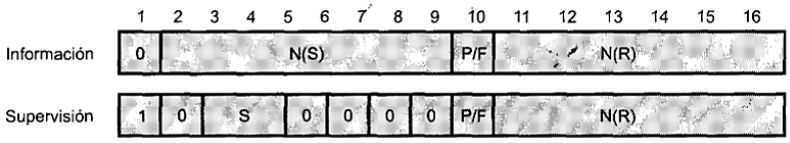
En este se definen tres tipos de tramas cada una de ellas con un formato diferente para el campo de control.

- **Tramas de información**. Transportan los datos generados por el usuario. Además, se incluye la información para control ARQ de errores y flujo.
- **Tramas de supervisión**. Proporcionan el mecanismo ARQ cuando la incorporación de las confirmaciones en las tramas de información no es factible.
- **Tramas no numeradas**. Proporcionan funciones complementarias para controlar el enlace.

El primer o los dos primeros bits del campo de control se utilizan para identificar el tipo de trama, además se incluye el bit poll/final.



(c) Formato del campo de control de 8 bits



(d) Formato del campo de control de 16 bits

LLC ofrece tres tipos de servicios. El servicio con modo de conexión, los otros dos son sin conexión y sin conexión confirmado.

Desarrollo:

El código para realizar la práctica es el siguiente:

```java
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.io.*;

import org.jnetpcap.Pcap;
import org.jnetpcap.PcapAddr;
import org.jnetpcap.PcapIf;
import org.jnetpcap.packet.PcapPacket;
import org.jnetpcap.packet.PcapPacketHandler;
import org.jnetpcap.PcapBpfProgram;


public class Captura {

    /**
     * Main startup method
     *
     * @param args
     *          ignored
     */
   static int no=1;

   public static int ntrama(){
       return no++;
   }
    private static String asString(final byte[] mac) {
    final StringBuilder buf = new StringBuilder();
    for (byte b : mac) {
      if (buf.length() != 0) {
        buf.append(':');
      }
      if (b >= 0 && b < 16) {
        buf.append('0');
      }
```

```
      buf.append(Integer.toHexString((b < 0) ? b + 256 : b).toUpperCase());
    }

    return buf.toString();
  }

    public static void main(String[] args) {
            Pcap pcap=null;
               try{
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        List<PcapIf> alldevs = new ArrayList<PcapIf>(); // Will be filled with NICs
        StringBuilder errbuf = new StringBuilder(); // For any error msgs
                System.out.println("[0]-->Realizar captura de paquetes al vuelo");
                System.out.println("[1]-->Cargar traza de captura desde archivo");
                System.out.print("\nElige una de las opciones:");
                int opcion = Integer.parseInt(br.readLine());
                if (opcion==1){

                    ////////////////////////lee archivo/////////////////////////

                //String fname = "archivo.pcap";
                String fname = "paquetes3.pcap";
                pcap = Pcap.openOffline(fname, errbuf);
                if (pcap == null) {
                  System.err.printf("Error while opening device for capture: "+ errbuf.toString());
                    return;
                 }//if
                } else if(opcion==0){
        /*************************************************************************
         * First get a list of devices on this system
         ************************************************************************/
        int r = Pcap.findAllDevs(alldevs, errbuf);
        if (r == Pcap.NOT_OK || alldevs.isEmpty()) {
            System.err.printf("Can't read list of devices, error is %s", errbuf
                .toString());
            return;
        }

        System.out.println("Network devices found:");

        int i = 0;
        for (PcapIf device : alldevs) {
            String description =
                (device.getDescription() != null) ? device.getDescription()
                    : "No description available";
                        final byte[] mac = device.getHardwareAddress();
            String dir_mac = (mac==null)?"No tiene direccion MAC":asString(mac);
                        System.out.printf("#%d: %s [%s] MAC:[%s]\n", i++, device.getName(), description, dir_mac);
                        List<PcapAddr> direcciones = device.getAddresses();
                        for(PcapAddr direccion:direcciones){
                            System.out.println(direccion.getAddr().toString());
                        }//foreach

```

```
88.         }//for
89.
90.                  System.out.print("\nEscribe el número de interfaz a utilizar:");
91.                  int interfaz = Integer.parseInt(br.readLine());
92.         PcapIf device = alldevs.get(interfaz); // We know we have atleast 1 device
93.         System.out
94.             .printf("\nChoosing '%s' on your behalf:\n",
95.                 (device.getDescription() != null) ? device.getDescription()
96.                     : device.getName());
97.
98.         /*************************************************************************
    **
99.          * Second we open up the selected device
100.                 ***********************************************************
    *********/
101.                        /*"snaplen" is short for 'snapshot length', as it refers t
    o the amount of actual data captured from each packet passing through the specifie
    d network interface.
102.                        64*1024 = 65536 bytes; campo len en Ethernet(16 bits) tam
    máx de trama */
103.
104.                int snaplen = 64 * 1024;           // Capture all packets, no truc
    ation
105.                int flags = Pcap.MODE_PROMISCUOUS; // capture all packets
106.                int timeout = 10 * 1000;           // 10 seconds in millis
107.
108.
109.                        pcap = Pcap.openLive(device.getName(), snaplen, flags, tim
    eout, errbuf);
110.
111.                if (pcap == null) {
112.                    System.err.printf("Error while opening device for capture: "
113.                        + errbuf.toString());
114.                    return;
115.                }//if
116.
117.                                /********F I L T R O********/
118.                    PcapBpfProgram filter = new PcapBpfProgram();
119.                    String expression =""; // "port 80";
120.                    int optimize = 0; // 1 means true, 0 means false
121.                    int netmask = 0;
122.                    int r2 = pcap.compile(filter, expression, optimize, netmask);
123.                    if (r2 != Pcap.OK) {
124.                        System.out.println("Filter error: " + pcap.getErr());
125.                    }//if
126.                    pcap.setFilter(filter);
127.                        /***************/
128.                    }//else if
129.
130.                /*************************************************************
    **********
131.                 * Third we create a packet handler which will receive packets fro
    m the
132.                 * libpcap loop.
133.                 ***********************************************************
    *****/
134.                PcapPacketHandler<String> jpacketHandler = new PcapPacketHandler<S
    tring>() {
135.                    public void nextPacket(PcapPacket packet, String user) {
```

```
                        System.out.printf("\n\n Paquete recibido el %s caplen=%-4d longitud=%-4d %s\n\n",
                                new Date(packet.getCaptureHeader().timestampInMillis()),
                                packet.getCaptureHeader().caplen(),  // Length actually captured
                                packet.getCaptureHeader().wirelen(), // Original length
                                user                                 // User supplied object
                                );


                                        /******Desencapsulado********/
                                        for(int i=0;i<packet.size();i++){
                                        System.out.printf("%02X ",packet.getUByte(i));

                                        if(i%16==15)
                                            System.out.println("");
                                        }//if

                                        System.out.println("\n\nEncabezado: "+ packet.toHexdump());

                                        int longitud = (packet.getUByte(12)*256)+packet.getUByte(13);
                                        int trama=1;
                                        if(longitud<=1500){
                                            int PFBit;
                                            System.out.printf("\n--->Trama IEEE802.3 %d\n",ntrama());
                                            System.out.printf("\nTamaño: %d bytes, Valor: %X\n",longitud,longitud );
                                            System.out.printf("|-->MAC Destino: %02X:%02X:%02X:%02X:%02X:%02X",packet.getUByte(0),packet.getUByte(1),packet.getUByte(2),packet.getUByte(3),packet.getUByte(4),packet.getUByte(5));
                                            System.out.printf("\n|-->MAC Origen: %02X:%02X:%02X:%02X:%02X:%02X",packet.getUByte(6),packet.getUByte(7),packet.getUByte(8),packet.getUByte(9),packet.getUByte(10),packet.getUByte(11));
                                            //System.out.printf("\n |-->DSAP: %02X",packet.getUByte(14));
                                            //System.out.println(packet.getUByte(15)& 0x00000001);
                                            int ssap = packet.getUByte(15);
                                            int dsap=packet.getUByte(14);
                                            String i_g=((dsap&0x1)==1)?"Grupal":"Individual";
                                            String c_r = ((ssap&0x1)==1)?"Respuesta":"Comando";
                                            int CRBit=((ssap&0x1)==1)?1:0;
                                            System.out.printf("\n|-->SSAP: Valor: %02X   %s\n",packet.getUByte(15), c_r);
                                            System.out.printf("|-->DSAP: Valor: %02X   %s\n",packet.getUByte(14), i_g);
                                            int control=packet.getUByte(16);
                                            int extendido=packet.getUByte(17);

                                            if(extendido==0){
                                                PFBit=((control>>4)&0b1);
                                            }else{
```

```
                                                PFBit=extendido&0b1;
                                            }
                                            String tipo="";

                                            if((control&0b11)==0b11){
                                                System.out.printf("El control es u
    n tipo no numerado %08d\n",Integer.parseInt(Integer.toString(control,2)));
                                                tipo=tipoNoEnumerado(CRBit, contro
    l, tipo);
                                                System.out.printf("Tipo: %s\n", ti
    po);
                                                System.out.printf("P/F: %s=%d\n",
    POLL_FINAL[CRBit], PFBit);
                                            }else if((control&0b11)==0b01){
                                                if (extendido == 0) {
                                                    System.out.printf("El control
    es de tipo supervision %08d\n", Integer.parseInt(Integer.toString(control, 2)));
                                                    System.out.printf("El valor N(
    R)=%X\n", control>>5);
                                                } else {
                                                    System.out.printf("El control
    es de tipo Supervision %08d %08d\n", Integer.parseInt(Integer.toString(extendido,
    2)),Integer.parseInt(Integer.toString(control, 2)));
                                                    System.out.printf("El valor N(
    R)=%X\n", extendido>>1);
                                                }
                                                tipo=supervision(control, tipo);
                                                System.out.printf("Tipo: %s\n", ti
    po);
                                                System.out.printf("P/F: %s=%d\n",
    POLL_FINAL[CRBit], PFBit);
                                            }else if((control&0b11)==0b0){
                                                if (extendido == 0) {
                                                    System.out.printf("El control
    es de tipo informacion %08d\n", Integer.parseInt(Integer.toString(control, 2)));
                                                    System.out.printf("El valor N(
    R)=%X\n", control>>5);
                                                    System.out.printf("El valor N(
    S)=%X\n", (control>>1)&0b111);
                                                } else {
                                                    System.out.printf("El control
    es de tipo informacion %08d %08d\n",Integer.parseInt(Integer.toString(extendido, 2
    )),Integer.parseInt(Integer.toString(control, 2)));
                                                    System.out.printf("El valor N(
    R)=%X\n", extendido>>1);
                                                    System.out.printf("El valor N(
    S)=%X\n", control>>1);
                                                }
                                                System.out.printf("P/F: %s=%d\n",
    POLL_FINAL[CRBit], PFBit);
                                            }else
                                                System.out.println("Error");

                                            System.out.println("");

                                        } else {
                                            //System.out.println("--
    >Trama ETHERNET numero-->"+cont.getCont());
                                        }
                    }
                };
```

```
218. 
219. 
220.                 /*************************************************************
    **********
221.                  * Fourth we enter the loop and tell it to capture 10 packets. The
     loop
222.                  * method does a mapping of pcap.datalink() DLT value to JProtocol
     ID, which
223.                  * is needed by JScanner. The scanner scans the packet buffer and
    decodes
224.                  * the headers. The mapping is done automatically, although a vari
    ation on
225.                  * the loop method exists that allows the programmer to sepecify e
    xactly
226.                  * which protocol ID to use as the data link type for this pcap in
    terface.
227.                  *************************************************************
    *********/
228.                 pcap.loop(-1, jpacketHandler, " ");
229. 
230.                 /*************************************************************
    **********
231.                  * Last thing to do is close the pcap handle
232.                  *************************************************************
    *********/
233.                 pcap.close();
234.                         }catch(IOException e){e.printStackTrace();}
235.             }
236. 
237.                 public static String supervision(int control, String tipo){
238.                     if(((control>>2)&0b11)==RR){
239.                         return "RR";
240.                     }else if(((control>>2)&0b11)==REJ){
241.                         return "REJ";
242.                     }else if(((control>>2)&0b11)==RNR){
243.                         return "RNR";
244.                     }else if(((control>>2)&0b11)==SREJ){
245.                         return "SREJ";
246.                     }else
247.                         return "";
248.                 }
249. 
250.                 public static String tipoNoEnumerado(int CRBit, int control, Strin
    g tipo){
251.                     if(CRBit==0){
252.                         if(((control^SNRM)==0) || ((control^SNRM)==16)){
253.                             tipo="SNRM";
254.                         }else if(((control^SNRME)==0) || ((control^SNRME)==16)){
255.                             tipo="SNRME";
256.                         }else if(((control^SARM_DM)==0) || ((control^SARM_DM)==16)
    ){
257.                             tipo="SARM_DM";
258.                         }else if(((control^SABM)==0) || ((control^SABM)==16)){
259.                             tipo="SABM";
260.                         }else if(((control^SABME)==0) || ((control^SABME)==16)){
261.                             tipo="SABME";
262.                         }else if(((control^UI)==0) || ((control^UI)==16)){
263.                             tipo="UI";
264.                         }else if(((control^DISC_RD)==0) || ((control^DISC_RD)==16)
    ){
265.                             tipo="DISC_RD";
```

```
266.                        }else if(((control^RSET)==0) || ((control^RSET)==16)){
267.                            tipo="RSET";
268.                        }else if(((control^XID)==0) || ((control^XID)==16)){
269.                            tipo="XID";
270.                        }
271.                    }else{
272.                        if ((control^SARM_DM)==0 || (control^SARM_DM)==16){
273.                            tipo="DM";
274.                        }else if ((control^UI)==0 || (control^UI)==16){
275.                            tipo="UI";
276.                        }else if ((control^UA)==0 || (control^UA)==16){
277.                            tipo="UA";
278.                        }else if ((control^DISC_RD)==0 || (control^DISC_RD)==16){

279.                            tipo="RD";
280.                        }else if ((control ^ XID) == 0 || (control ^ XID) == 16){

281.                            tipo = "XID";
282.                        }
283.                    }
284.                    return tipo;
285.                }
286.
287.                static final String[] POLL_FINAL=new String[]{"P","F"};
288.
289.                static final int SNRM=0b10000011;
290.                static final int SNRME=0b11001111;
291.                static final int SARM_DM=0b00001111;
292.                static final int SABM=0b00101111;
293.                static final int SABME=0b01101111;
294.                static final int UI=0b00000011;
295.                static final int UA=0b01100011;
296.                static final int DISC_RD=0b01000011;
297.                static final int RSET=0b10001111;
298.                static final int XID=0b10101111;
299.
300.                static final int RR=0b00;
301.                static final int REJ=0b10;
302.                static final int RNR=0b01;
303.                static final int SREJ=0b11;
304.        }
```

Pruebas:

A continuación, se hacen pruebas donde a través de un archivo. pcap, se nos facilitan algunas tramas para su análisis.

```
run:
[0]-->Realizar captura de paquetes al vuelo
[1]-->Cargar traza de captura desde archivo

Elige una de las opciones:1
```

Trama 1:

```
--->Trama IEEE802.3 1

Tamaño: 3 bytes, Valor: 3
|-->MAC Destino: 00:02:B3:9C:AE:BA
|-->MAC Origen: 00:02:B3:9C:DF:1B
|-->SSAP: Valor: F0   Comando
|-->DSAP: Valor: F0   Individual
El control es un tipo no numerado 01111111
Tipo: SABME
P/F: P=1
```

Trama 2:

```
--->Trama IEEE802.3 2

Tamaño: 3 bytes, Valor: 3
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->SSAP: Valor: F1   Respuesta
|-->DSAP: Valor: F0   Individual
El control es un tipo no numerado 01110011
Tipo: UA
P/F: F=1
```

Trama 11:

```
--->Trama IEEE802.3 11

Tamaño: 95 bytes, Valor: 5F
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->SSAP: Valor: F0   Comando
|-->DSAP: Valor: F0   Individual
Error
```

Trama 16:

```
--->Trama IEEE802.3 16

Tamaño: 4 bytes, Valor: 4
|-->MAC Destino: 00:02:B3:9C:AE:BA
|-->MAC Origen: 00:02:B3:9C:DF:1B
|-->SSAP: Valor: F1   Respuesta
|-->DSAP: Valor: F0   Individual
El control es de tipo Supervision 00000110 00000001
El valor N(R)=3
Tipo: RR
P/F: F=0
```

Trama 24:

```
--->Trama IEEE802.3 24

Tamaño: 4 bytes, Valor: 4
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->SSAP: Valor: F1   Respuesta
|-->DSAP: Valor: F0   Individual
El control es de tipo Supervision 00001101 00000001
El valor N(R)=6
Tipo: RR
P/F: F=1
```

Trama 29:

```
--->Trama IEEE802.3 29

Tamaño: 4 bytes, Valor: 4
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->SSAP: Valor: F1   Respuesta
|-->DSAP: Valor: F0   Individual
El control es de tipo Supervision 00010001 00000001
El valor N(R)=8
Tipo: RR
P/F: F=1
```

Conclusiones.

Mario Alberto Miranda Sandoval.

En esta práctica comprendí mejor el análisis de tramas, ya que este es importante para poder desarrollar este programa, por lo que sin el conocimiento no se hubiera podido haber realizado, posteriormente comprendí de mejor manera lo que es el protocolo LLC, además, de saber que no es el único que existe.

Rojas Alvarado Luis Enrique.

En ésta práctica pudimos observar el comportamiento de una trama identificando las diferentes partes que la forman en su encabezado de IP, por lo que usamos el programa proporcionado por el profesor desde su servidor simplemente para hacer un simple análisis de las tramas.