



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Práctica 2: Análisis de tramas IEEE 802.3

Unidad de aprendizaje: Redes de computadoras

Grupo: 2CM9

Alumnos:

- López Cedillo Alexander
- Ontiveros Salazar Alan Enrique
- Rojas Espinoza Omar

Profesor:

Moreno Cervantes Axel Ernesto

20 de noviembre de 2017

Práctica 2: Análisis de tramas IEEE 802.3

2CM9

20 de noviembre de 2017

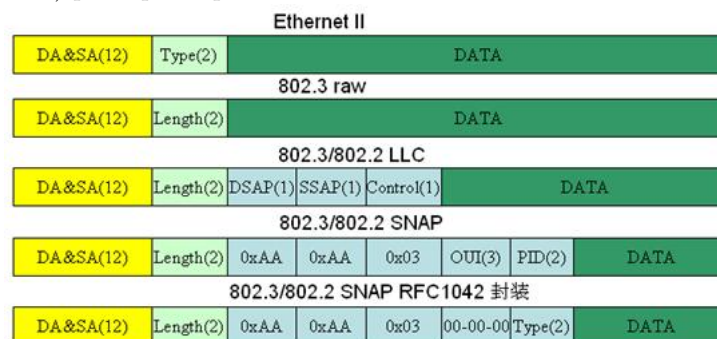
1. Marco Teórico

1.1. Formato de una trama IEEE 802.3

Los campos que conforman una trama IEEE 802.3 son los siguientes:

- Preámbulo: el patrón de unos y ceros alternados les indica a las estaciones receptoras que una trama es Ethernet o IEEE 802.3. La trama Ethernet incluye un byte adicional que es el equivalente al campo Inicio de trama (SOF) de la trama IEEE 802.3.
- Inicio de trama (SOF): el byte delimitador de IEEE 802.3 finaliza con dos bits 1 consecutivos, que sirven para sincronizar las porciones de recepción de trama de todas las estaciones de la LAN. SOF se especifica explícitamente en Ethernet.
- Direcciones destino y origen: vienen determinadas por las direcciones MAC únicas de cada tarjeta de red (6 bytes en hexadecimal). Los primeros 3 bytes de las direcciones son especificados por IEEE según el proveedor o fabricante. El proveedor de Ethernet o IEEE 802.3 especifica los últimos 3 bytes. La dirección origen siempre es una dirección de broadcast única (de nodo único). La dirección destino puede ser de broadcast única, de broadcast múltiple (grupo) o de broadcast (todos los nodos).
- Longitud (IEEE 802.3): la longitud indica la cantidad de bytes de datos que sigue este campo.
- Datos (Ethernet): una vez que se ha completado el procesamiento de la capa física y de la capa de enlace, los datos contenidos en la trama se envían a un protocolo de capa superior, que se identifica en el campo tipo. Aunque la versión 2 de Ethernet no especifica ningún relleno, al contrario de lo que sucede con IEEE 802.3, Ethernet espera por lo menos 46 bytes de datos.
- Secuencia de verificación de trama (FCS): esta secuencia contiene un valor de verificación CRC (Control de Redundancia Cíclica) de 4 bytes, creado por el dispositivo emisor y recalculado por el dispositivo receptor para verificar la existencia de tramas dañadas.

Cuando un paquete es recibido por el destinatario adecuado, les retira la cabecera de Ethernet y el checksum de verificación de la trama, comprueba que los datos corresponden a un mensaje IP y entonces lo pasa a dicho protocolo (capa de red-Internet) para que lo procese.



Un punto de acceso de servicio (SAP) es una etiqueta de identificación para los puntos finales de red utilizados en la red de interconexión de sistemas abiertos (OSI). El estándar IEEE incluye esta subcapa que añade las etiquetas estándar de 8-bit DSAP (Destination Service Access Point) y SSAP (Source Service Access Point) a los paquetes del tipo de conexión. También usado en funciones auxiliares como Control de flujo. Hay sitio para 64 números SAP globalmente asignados, y la IEEE no los asigna a la ligera. IP no tiene un número SAP asignado, porque solo los “estándares internacionales” pueden tener números SAP. Los protocolos que no lo son pueden usar un número SAP del espacio de SAP administrado localmente.

HDLC (High-Level Data Link Control, control de enlace de datos de alto nivel) es un protocolo de comunicaciones de propósito general punto a punto, que opera a nivel de enlace de datos. Proporciona recuperación de errores en caso de pérdida de paquetes de datos, fallos de secuencia y otros, por lo que ofrece una comunicación confiable entre el transmisor y el receptor.

En este protocolo podemos encontrar tres tipos de tramas: Tramas que transportan los datos del usuario, las tramas de supervisión y las tramas de gestión (o tramas no numeradas).

1.2. Tramas de Supervisión

Las tramas de supervisión se utilizan para el reconocimiento de tramas, control de flujo y control de errores (siempre que no sea posible hacerlo mediante las tramas de información).

Las órdenes que podemos encontrar en las tramas de gestión son las siguientes:

- Campos.
- DSAP
- SSAP.

00001	SNRM	
11011	SNRME	
11000	SARM	DM
11010	SARME	
11100	SABM	
11110	SABME	
00000	UI	UI
00110		UA
00010	DISC	RD
10000	SIM	RIM
00100	UP	

2. Desarrollo

Durante la practica se realizó el análisis de tramas IEEE 802.3 capturadas a través de la librería PCAP.

Primeramente revisamos los bytes 13 y 14 de la trama para verificar si se trata de una trama IEEE 802.3 o de una trama Ethernet e imprimimos el valor obtenido.

```
int longitud = (uInt(trama[12]) << 8) | uInt(trama[13]);
System.out.printf("\nLongitud: %d (%04X)", longitud, longitud);
```

En caso de obtener una trama IEEE 802.3 proseguimos a su análisis.

Primero, extraemos las direcciones MAC de origen y destino, y las mostramos en la terminal.

```

if (longitud < 1500) {
    System.out.println("---->Trama IEEE802.3");
    byte[] macOrigen = new byte[6];
    byte[] macDestino = new byte[6];
    System.arraycopy(trama, 6, macOrigen, 0, 6);
    System.arraycopy(trama, 0, macDestino, 0, 6);
    System.out.println(" |-->MAC Destino: " + macString(macDestino));
    System.out.println(" |-->MAC Origen: " + macString(macOrigen));
}

```

Procedemos a revisar el byte 15 de la trama para determinar el tipo de dsap. En caso de ser 0x00 sabemos que es de tipo individual, en otro caso de grupo. Verificamos el tipo e imprimimos en la terminal el valor del dsap y el tipo.

```

byte dsap = trama[14];
System.out.printf(" |-->DSAP: %02X ", dsap);
if ((dsap & 0x01) == 0x00) System.out.println("individual");
else System.out.println("grupo");

```

Posteriormente, extraemos el ssap y verificamos de qué tipo se trata, ya sea comando o respuesta. Imprimimos su valor y el tipo.

```

byte ssap = trama[15];
System.out.printf(" |-->SSAP: %02X ", ssap);
if ((ssap & 0x01) == 0x00) System.out.println("comando");
else System.out.println("respuesta");

```

Después revisamos de qué tipo de trama se trata, para ello revisamos el byte 17. Sabemos que si este valor es de 0x00 se trata de una trama I, si es de 0x01 nos encontramos con una trama S, y si vale 0x03 es una trama U.

Para la trama I, volvemos a revisar la longitud de la trama. Si es menor o igual a 3 se trata de una trama normal, en otro caso, nos encontramos con la versión extendida. Imprimimos tanto el tipo de trama como su versión.

En ambos casos calculamos los valores de N(S), N(R) y P/F según la versión de la trama. Para la trama normal utilizamos el byte 17, mientras que para la versión extendida utilizamos el 17 y el 18.

```

if ((trama[16] & 0x01) == 0) { //I frame
    System.out.println(" |-->Trama I");
    if (longitud <= 3) {
        System.out.println(" |-->1 byte (normal)");
        n_s = (trama[16] >> 1) & 0x07;
        p_f = (trama[16] >> 4) & 0x01;
        n_r = (trama[16] >> 5) & 0x07;
    } else {
        System.out.println(" |-->2 bytes (extendido)");
        n_s = (trama[16] >> 1) & 0x7F;
        p_f = trama[17] & 0x01;
        n_r = (trama[17] >> 1) & 0x7F;
    }
    System.out.println(" |-->N(R) = " + n_r + "\n |-->N(S) = " + n_s + "\n |-->P/F  
↪ = " + p_f);
}

```

Para las tramas de tipo S volvemos a revisar el campo de longitud de la trama original y verificamos si es mayor que 3. En caso afirmativo se calculan el N(R) y el P/F con los bytes 17 y 18. En el caso contrario sólo utilizamos el byte 17.

```

} else if ((trama[16] & 0x03) == 0x01) { //S frame
    System.out.println(" |-->Trama S");
}

```

```

if (longitud <= 3) {
    System.out.println(" |-->1 byte");
    codigo = (trama[16] >> 2) & 0x03;
    p_f = (trama[16] >> 4) & 0x01;
    n_r = (trama[16] >> 5) & 0x07;
} else {
    System.out.println(" |-->2 bytes");
    codigo = (trama[16] >> 2) & 0x03;
    p_f = trama[17] & 0x01;
    n_r = (trama[17] >> 1) & 0x7F;
}
System.out.println(" |-->Codigo = " + bin2str(codigo) + ": " + codigoS(codigo)
    ↪ + "\n |-->N(R) = " + n_r + "\n |-->P/F = " + p_f);

```

Finalmente, para las tramas tipo U calculamos el P/F con el byte 17 y obtenemos el código de ese mismo byte.

```

} else if ((trama[16] & 0x03) == 0x03) { //U frame
    System.out.println(" |-->Trama U");
    p_f = (trama[16] >> 4) & 0x01;
    codigo = ((trama[16] >> 3) & 0x1C) | ((trama[16] >> 2) & 0x03);
    System.out.println(" |-->Codigo = " + bin2str(codigo) + ": " + codigoU(codigo,
    ↪ ssap & 0x01) + "\n |-->P/F = " + p_f);
}

```

3. Pruebas

Capturando tramas desde el archivo de prueba:

```
C:\Windows\System32\cmd.exe
C:\Users\alan-\Documents\trabajos-escom\3º\redes\practica 2>java -cp .;jnetpcap.jar Captura
[0]-->Realizar captura de paquetes al vuelo
[1]-->Cargar traza de captura desde archivo

Elige una de las opciones:1

Paquete recibido el Mon Mar 24 12:24:33 CST 2014 caplen=64 longitud=64

00 02 B3 9C AE BA 00 02 B3 9C DF 1B 00 03 F0 F0
7F 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 43 05 90 6D

Longitud: 3 (0003)-->Trama IEEE802.3
|-->MAC Destino: 00:02:B3:9C:AE:BA
|-->MAC Origen: 00:02:B3:9C:DF:1B
|-->DSAP: F0 individual
|-->SSAP: F0 comando
|-->Trama U
|-->Codigo = 01111: SABME (Activacion de modo respuesta asincrona, extendida)
|-->P/F = 1

Paquete recibido el Mon Mar 24 12:24:33 CST 2014 caplen=64 longitud=64

00 02 B3 9C DF 1B 00 02 B3 9C AE BA 00 03 F0 F1
73 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 54 90 6D

Longitud: 3 (0003)-->Trama IEEE802.3
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->DSAP: F0 individual
|-->SSAP: F1 respuesta
|-->Trama U
|-->Codigo = 01100: UA (Reconocimiento sin numerar)
|-->P/F = 1
```

```

C:\Windows\System32\cmd.exe
Paquete recibido el Mon Mar 24 12:24:33 CST 2014 caplen=64 longitud=64

00 02 B3 9C DF 1B 00 02 B3 9C AE BA 00 04 F0 F1
01 13 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 7C 9B 6D

Longitud: 4 (0004)--->Trama IEEE802.3
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->DSAP: F0 individual
|-->SSAP: F1 respuesta
|-->Trama S
|-->2 bytes
|-->Codigo = 00000: RR (listo para recibir)
|-->N(R) = 9
|-->P/F = 1

Paquete recibido el Mon Mar 24 12:24:33 CST 2014 caplen=64 longitud=64

00 02 B3 9C AE BA 00 02 B3 9C DF 1B 00 03 F0 F0
53 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 CB 9B 6D

Longitud: 3 (0003)--->Trama IEEE802.3
|-->MAC Destino: 00:02:B3:9C:AE:BA
|-->MAC Origen: 00:02:B3:9C:DF:1B
|-->DSAP: F0 individual
|-->SSAP: F0 comando
|-->Trama U
|-->Codigo = 01000: DISC (Desconexion)
|-->P/F = 1

Paquete recibido el Mon Mar 24 12:24:33 CST 2014 caplen=64 longitud=64

00 02 B3 9C DF 1B 00 02 B3 9C AE BA 00 03 F0 F1
73 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 77 9C 6D

Longitud: 3 (0003)--->Trama IEEE802.3
|-->MAC Destino: 00:02:B3:9C:DF:1B
|-->MAC Origen: 00:02:B3:9C:AE:BA
|-->DSAP: F0 individual
|-->SSAP: F1 respuesta
|-->Trama U
|-->Codigo = 01100: UA (Reconocimiento sin numerar)
|-->P/F = 1

C:\Users\alan-\Documents\trabajos-escom\3º\redes\practica 2>

```

4. Conclusiones

- **López Cedillo Alexander:**

- **Ontiveros Salazar Alan Enrique:** En esta práctica se realizó correctamente el análisis de una trama IEEE 802.3, la cual es identificada por el campo Longitud si es menor a 1500 bytes.

Es menos potente que una trama Ethernet, pues esta última, en vez de almacenar un campo Longitud, almacena un campo Tipo que es capaz de identificar a más protocolos.

- **Rojas Espinoza Omar:**

Referencias