

*Instituto Politécnico Nacional*

*Escuela Superior de Cómputo*

*Redes de computadoras*

*Práctica 1*

*López Cedillo Alexander*

*Ontiveros Salazar Alan Enrique*

*Rojas Espinoza Omar*

*2CM9*

*Profesor: Moreno Cervantes Axel Ernesto*

**Introducción**

Una trama es la unidad de transmisión de las operaciones de la capa o nivel 2, enlace de datos, de una red. Cuando la capa de enlace de datos recibe un mensaje, le da formato para convertirlo en una trama de datos o paquete. Los campos que componen una trama ethernet son los siguientes:

* Preámbulo (Preamble)

Campo de 7 bytes de longitud con una secuencia de bits utilizada para sincronizar y estabilizar el medio físico antes de iniciar la transmisión. Es una secuencia de unos y ceros. El patrón es el siguiente:

               10101010 10101010 10101010 10101010 10101010 10101010 10101010

* SFD (Start Frame Delimiter)

Delimitador de inicio de trama. Campo de 1 byte de longitud que contiene la secuencia 10101011. Indica el inicio de una trama de datos.

* Dirección de destino (Destination Address)

Campo de 6 bytes de longitud que contiene la dirección MAC a la que se envía la trama. El bit más a la izquierda del campo indica cuando la dirección es individual (indicado por un 0) o un grupo de direcciones (indicado por un 1). El segundo bit desde la izquierda indica cuando la dirección destino es globalmente administrada (indicado por un 0).

La capa de enlace de datos del remitente añade la dirección de destino a la trama. La capa de enlace de datos del destinatario examina la dirección de destino para identificar los mensajes a recibir.

* Dirección de origen (Source Address). Campo de 6 bytes de longitud que contiene la dirección MAC del dispositivo que envía la trama. La dirección de origen es siempre una dirección individual y el bit más a la izquierda es siempre 0. Con ella el receptor conoce a quien debe dirigir las respuestas del mensaje.
* Tipo de protocolo o longitud. Campo de 2 bytes de longitud. Este campo es el que distingue a las tramas 802.3 de las tramas Ethernet.

Valores para este campo iguales o menores de x05DC (1500 en decimal) indican que es una trama 802.3 y el valor representa la longitud del campo de datos.

Valores para este campo iguales o mayores de x0600 indican que es una trama Ethernet y el valor representa el tipo de protocolo.

* Datos (Payload). Campo de 46 a 1500 bytes de longitud. Contiene los datos a transferir entre origen y destino. Si este campo fuera menor de 46 bytes se añade un campo de ‘relleno’ para mantener el tamaño mínimo de paquete.

* FCS (Frame Check Sequence). Secuencia de verificación de trama.

Campo de 4 bytes de longitud que contiene un valor de para control de errores, CRC (Cyclical Redundancy Check). La verificación de redundancia cíclica (CRC), consiste en un valor calculado por el emisor que resume todos los datos de la trama. El receptor calcula nuevamente el valor y, si coincide con el de la trama, entiende que la trama se ha transmitido sin errores. El campo FCS es generado sobre los campos dirección de destino, la dirección de origen, el tipo/longitud y datos.

**Desarrollo**

La práctica consiste en capturar tramas que pasan a través de la tarjeta de red configurada en modo promiscuo. A partir de estas tramas analizamos si se trataba de una trama Ethernet, y posteriormente si el protocolo de Internet era IP. Para ello utilizamos la librería pcap y el código que nos fue proporcionado por el profesor.

Para esta práctica supusimos que la trama con la que se trataba era Ethernet, por lo cual no se revisó que el campo tipo/longitud fuera mayor o igual a 1500 bytes.

Posteriormente, revisamos el byte 13 y 14 de la trama para verificar que el protocolo que se iba a utilizar era IPv4, es decir, revisamos que el valor fuera igual a 0x08 en el byte 13 y 0x00en el byte 14 (línea 176 del código).

Una vez validado el tipo de protocolo como IPv4 continuamos a calcular el checksum para verificar posibles errores y dar por buena la trama.

El procedimiento que seguimos fue el siguiente (líneas 178-187):

1. Calcular la longitud del encabezado IP.
2. Crear un nuevo arreglo de bytes para almacenar el encabezado.
3. Identificar IP de origen e IP destino.
4. Calcular el checksum utilizando el método proporcionado por el profesor.

Finalmente, procedimos a identificar el protocolo que se utilizó en la capa de transporte (línea 188 y 209). Para esto revisamos el valor del byte 24 de la trama. Si el valor resultaba ser 0x06, sabíamos que se trataba de TCP. Por otro lado, si el valor era 0x11, el protocolo era UDP.

Para ambos casos se requería calcular el valor del checksum. Sin embargo, para ambos había que calcular previamente un pseudo-header que se utiliza en el cálculo del checksum.

En cuanto al checksum de TCP, lo calculamos de la manera siguiente (líneas 188-208):

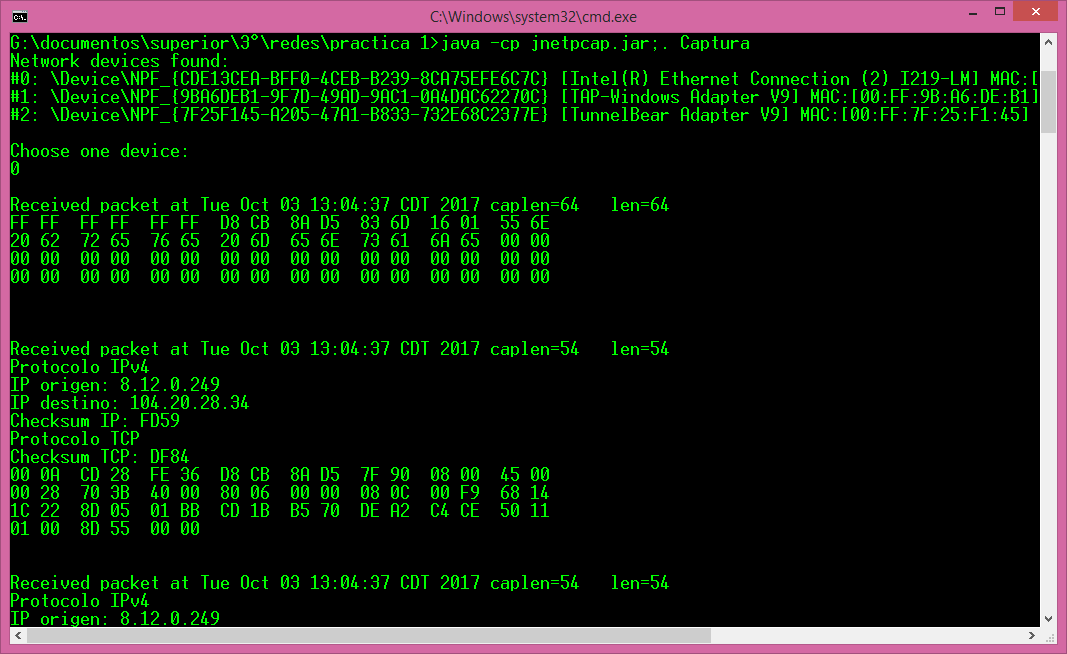
1. Calculamos la longitud del encabezado TCP.
2. Creamos un nuevo arreglo que almacenaba el encabezado TCP.
3. Creamos un arreglo que iba a contener la información correspondiente al pseudo-header.
4. Agregamos la IP de origen y destino al pseudo-header, en ese orden.
5. Agregamos la información correspondiente a los bytes 9 y 10. En el byte 9 se asigna el valor de 0 y en el byte 10 el valor de 0x06 correspondiente a que se trata de un protocolo TCP.
6. Calculamos la longitud del payload utilizando la longitud total menos la longitud del encabezado IP menos 14 que es la longitud del encabezado TCP.
7. Agregamos la información del payload a los bytes 11 y 12.
8. Creamos un nuevo arreglo que contendrá el payload de TCP y copiamos la información.
9. Creamos un arreglo auxiliar que contendrá: el pseudo-header, el encabezado TCP y el payload.
10. Calculamos el checksum utilizando el arreglo auxiliar.

Por otra parte el checksum de UPD, lo calculamos de la manera siguiente (líneas 210-227):

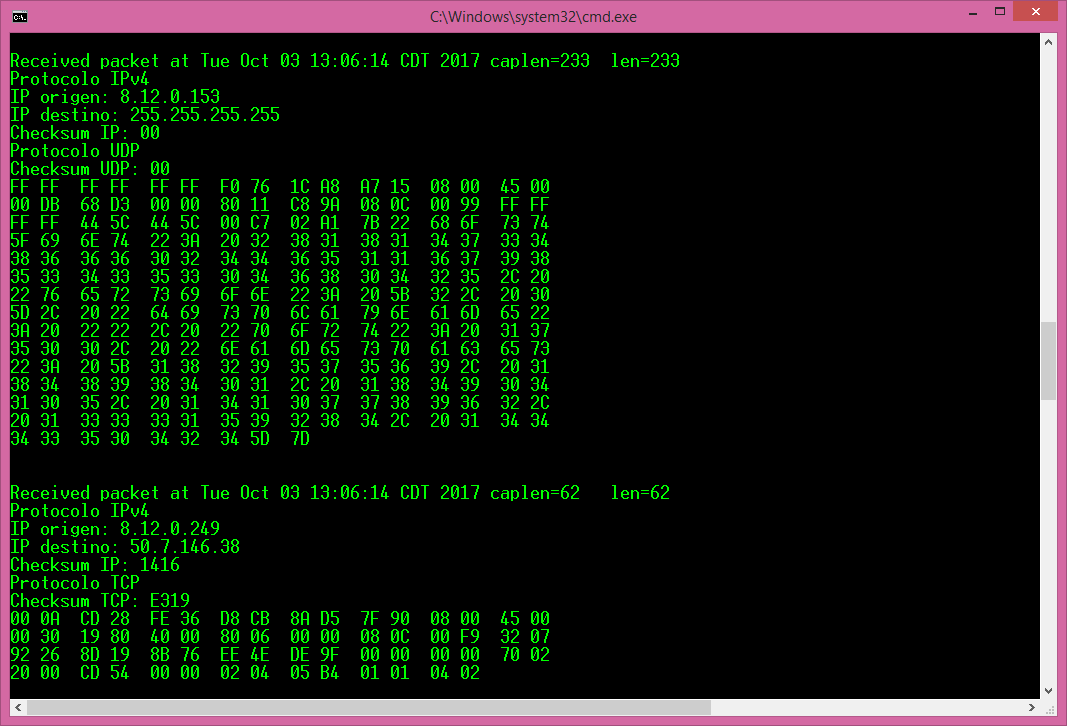
1. Creamos un arreglo que contendrá el encabezado UDP y copiamos la información.
2. Creamos un arreglo de 12 bytes que nos servirá para crear el pseudo-header.
3. Copiamos el IP origen y el IP destino al pseudo-header.
4. Inicializamos los bytes 9 y 10. El 9 se inicializa a 0x00 y el 10 a 0x11 debido a que se trata del protocolo UDP.
5. Copiamos el encabezado UDP al pseudo-header.
6. Calculamos la longitud del payload: longitud de la trama menos longitud de IP menos longitud de UDP menos 14.
7. Creamos un arreglo de bytes que contendrá el payload y copiamos la información de la trama.
8. Creamos un arreglo temporal de bytes para realizar el cálculo del checksum. Agregamos: pseudo-header, encabezado UDP y el payload, en ese orden.
9. Realizamos el cálculo del checksum.

**Capturas**

1. Corriendo el programa:



1. Capturando tramas con TCP y UDP:



**Conclusiones**

* López Cedillo Alexander:

La práctica fue realmente enriquecedora en cuanto a los temas del curso, ya que me ayudo a entender mejor sobre la estructura de las tramas Ethernet. También por otro lado, me proporciono claridad sobre el principio de encapsulación que se debe de cumplir entre cada una de las capas del modelo OSI. Por último, fue interesante el tema de manipulación de la trama como tal para aislar el encabezado de los datos, así como para generar el pseudo-header necesario para la validación del PDU TCP y UDP.

* Ontiveros Salazar Alan Enrique:

Es esta práctica implementamos el algoritmo de Checksum aplicado a verificar tramas Ethernet. Comprobamos que dichas tramas fueran IPv4 verificando que los bytes 13 y 14 sean iguales a 0x0800, para posteriormente decidir si el protocolo de la capa de transporte era TCP o UDP. Finalmente, comprobamos el campo checksum de estos protocolos.

Al correr el programa, verificamos que varias tramas de TCP llegaban incompletas, por lo que el checksum era diferente de cero; mientras que las tramas UDP tendían a llegar sin errores la mayoría del tiempo.

* Rojas Espinoza Omar:

Lo más importante de esta práctica, es la utilización del algoritmo de checksum para identificar los errores en una trama, la cual, asumimos inicialmente, que es una trama Ethernet. El checksum se validó con base en el protocolo IPv4 y una vez que se ha dado por buena una trama, posteriormente verificamos si el protocolo para el transporte es TCP o UDP, en lo que se utilizó el valor calculado del checksum, en el que intervenía un pseudo-header.